

# EMSI – IIR – INFORMATIQUE – DEVELOPPEMENT



## ASP.NET

Projet WEB 1 - MicroSoft **ASP.Net**  
**Prof. Y. EL-GHAYAM**



Par :



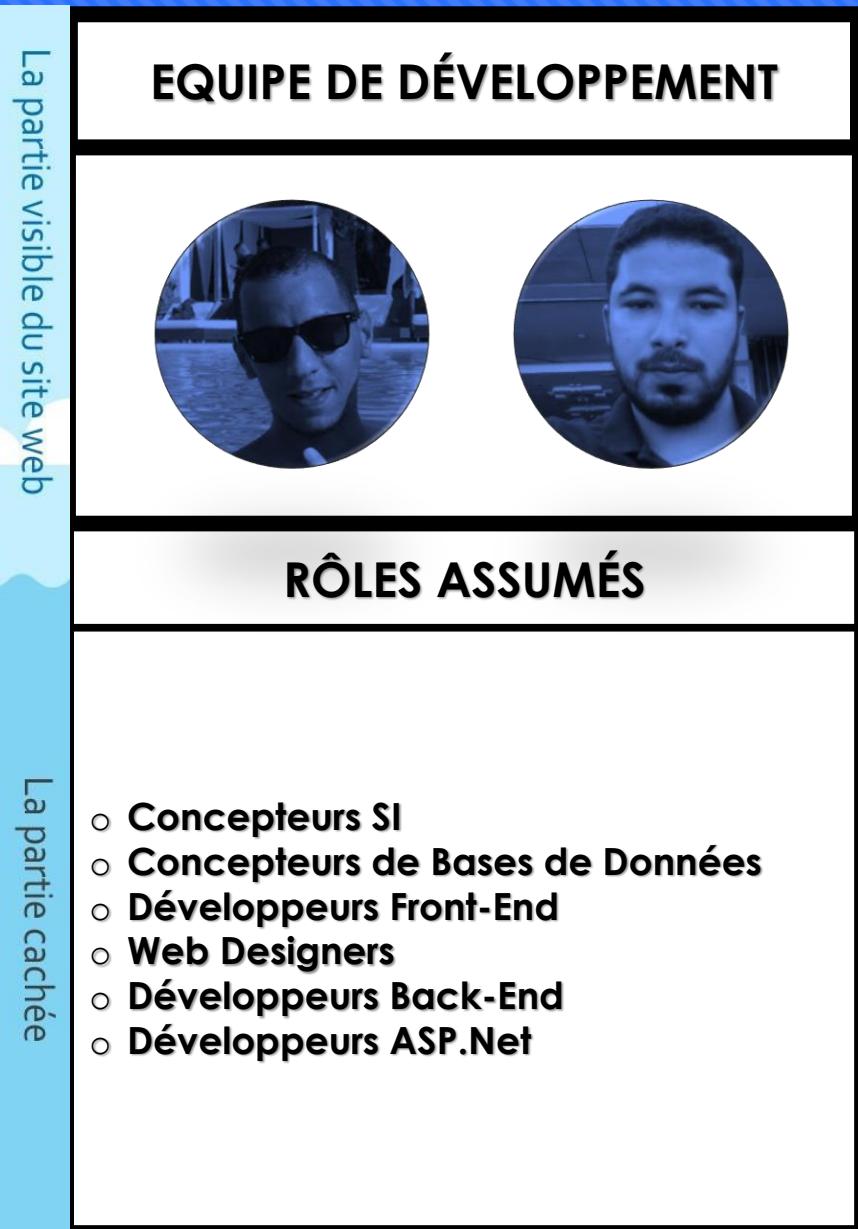
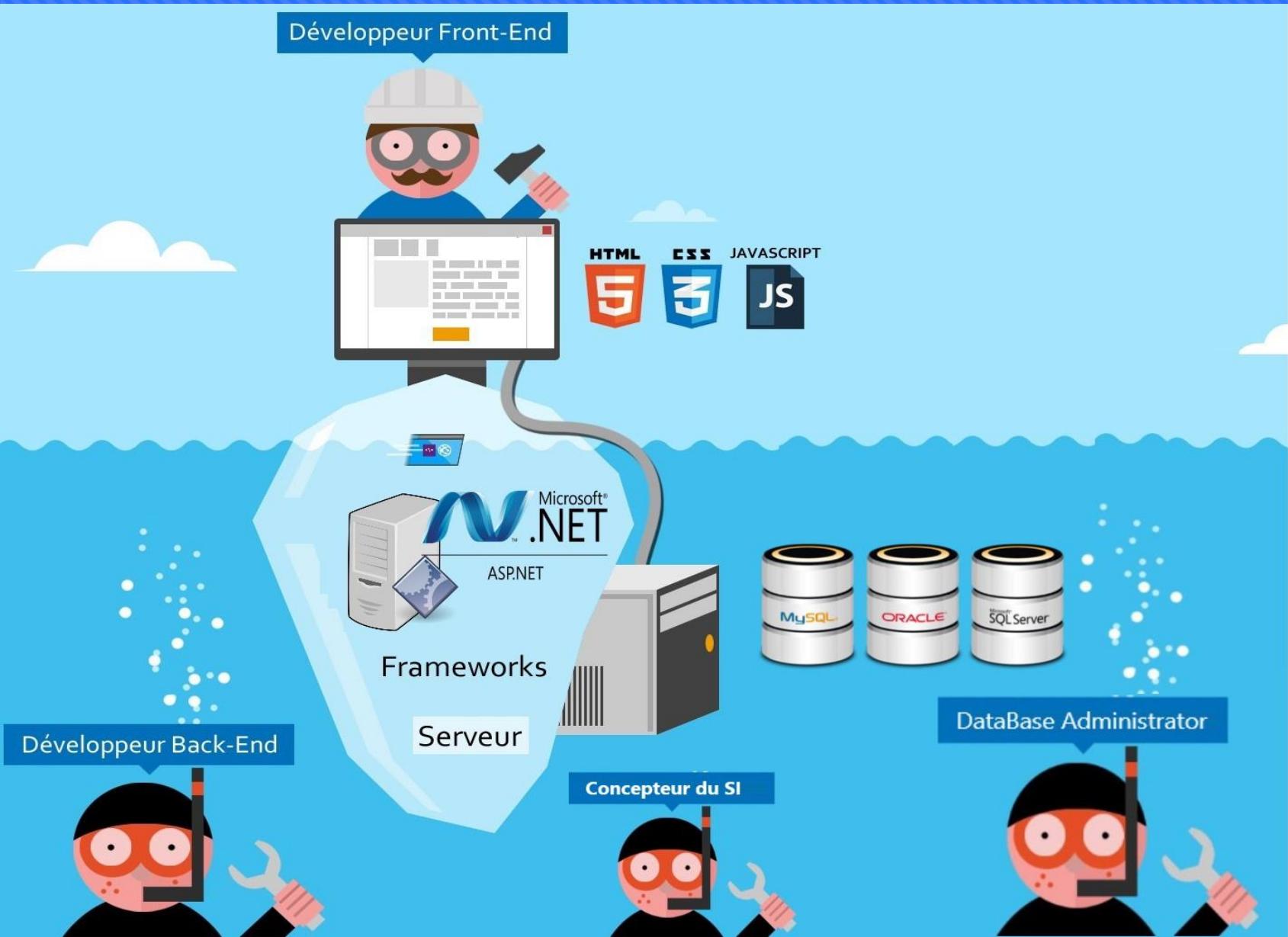
Hamza **OUDADDA**

&



Youssef **MAHTAT**

# Présentation du Contexte de Développement :



# LIVRABLES :

## ETUDES DU PROJET

Réalité et Besoins du Projet, et les Etudes et analyse de l'équipe développement.



### PLAN :

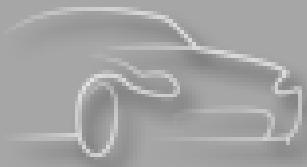
- PROMOTING
- Chartes et Consignes du projet.
- Plan du projet.
- Etude préliminaire et Etude de l'existant.
- Etude Fonctionnelle.
- Etudes Techniques.
- Conception et Modélisation.
- CONCEPTS ET ALGORITHME.
- ARCHTECTURE & ASP.NET APPLY DESIGN.



## PROMOTION DE L'APPLY :

Ste

OUDADDA & MAHTAT **TaxiTrafic**



O&M  
*TaxiTrafic*

Le TAXI 2.0, L'innovation c'est nous !!





# CHARTES ET CONSIGNES DU PROJET

# Description du Projet :

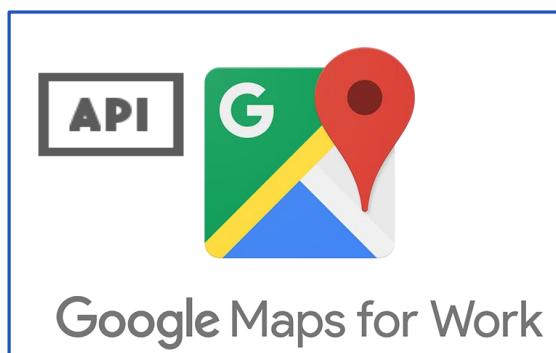
Référence du projet	Projet EMSI 4°IIR2 – DotNet – N° 1
Intitulé du projet	Projet de Réalisation d'une Application WEB DOTNET ( <b>ASP.NET</b> ), sous Microsoft.NET, pour la gestion des taxis d'une Ste de transport.
Population cible	Les Gestionnaires de la Société de transport (Administrateurs) et Chauffeurs de Taxis.
Membres Réalisateur, Concepteurs et Chercheurs	Hamza <b>OUDADDA</b> & Youssef <b>MAHTAT</b>
Partenaires Externes	Professeur <b>Mr. EL-GHAYAM</b> , Etudiants EMSI ou d'autres partenaires

- Charte du Projet – Consigne et présentation de la réalité des besoins

## CONSIGNES ET BESOINS (CONTEXTE) DU PROJET (PAR MR. EL-GHAYAM ) :

- Une société de transport voudrait automatiser la tâche de la répartition des clients sur l'ensemble des taxis disponibles :
  - Pour tout taxi, on devra lui affecter un maximum des clients avant de passer au deuxième taxi.
  - Un taxi peut démarrer le travail de 7h du matin jusqu'à 20h du soir.
  - Un taxi ne peut avoir qu'un seul client à un instant t.
  - L'affectation des clients devra respecter la succession dans le temps et la succession géographique.
  - La marge de retard tolérée de chaque rendez-vous est de 5min.
- Exemple : pour l'exemple suivant, on affecte au même chauffeur les clients suivants :
  - *Client\_1 = {heure\_départ = 7 : 00, adresse\_depar = Rabat, adresse\_arrivée = Casa, ...}*
  - *Client\_2 = {heure\_depart = 09 : 00, adresse\_depart = casa, ...}*
- Certains clients peuvent faire plusieurs courses dans la même semaine et d'une façon périodique (chaque semaine). La périodicité peut être limitée par une date ou non.
- Un Client appelle pour réserver une course.
- Les informations des clients sont supposées enregistrées d'avance. Les informations manipulées d'un client sont: le nom complet, l'heure de départ, l'adresse de départ, l'adresse d'arrivée et l'état périodique (oui/non), la date limite de la périodicité au cas où c'est limitée.
- Un client peut avoir plusieurs adresses de départs et d'arrivées et pour chacune une heure de départ différente.
- Les chauffeurs peuvent consulter en tout moment leurs tableaux des affectations. Chacun d'eux ne peut consulter que son propre affectation des courses.

# API à Utilisés :



- API Google Maps (**DistanceMatrix**), pour le calculer une estimation de la distance en terme de kilométrage et en terme de temps (en seconde) :

- [https://maps.googleapis.com/maps/api/distancematrix/xml?origins=\\$origin&destinations=\\$dest&mode=driving&region=french](https://maps.googleapis.com/maps/api/distancematrix/xml?origins=$origin&destinations=$dest&mode=driving&region=french)

La chaîne indiquant l'adresse d'origine

La chaîne indiquant l'adresse de destination

# Objectifs du Projet :

(Objectifs Directs Partiels)



- **Programmer** les modules et interfaces en POO et Programmation orientée évènements.
- **Rendre** Les fonctionnalités de la plateforme modulables en modules et bibliothèques pour faciliter la MàJ, pouvoir les intégrer dans d'autres projets ou avoir la possibilité de les isoler.
- Automatiser les processus de planification et répartition des trafics de taxis.
- Faciliter les opérations liées à la manipulation des courses de Taxis.
- **Elaborer** des Codes et Algorithmes claires.
- **Réaliser** une Conception adéquate et fonctionnelle.
- **Editer** des Interfaces esthétiques, graphiquement jolies et donnent envie à l'utilisateur.
- **Ajouter** les concepts : Authentification & Sessions ainsi que les Service de recherche et filtrage.
- **Permettre** aux **administrators** de superviser la plateforme via des interfaces dédiées.

# Facteurs Clés de succès :

- L'implication des membres de l'équipe, consultants et partenaires projet.
- La bonne formation sur les technologies et outils (ASP.Net, C#.Net, ...).
- La bonne approche de la réalité des taxis et les courses de taxis.
- La Modélisation et Conception complet et flexible de la réalité du projet
- La haute disponibilité de l'environnement de déploiement.

# Risques et Contraintes du Projet :

- Gestion du Projet
- Approche métier (difficulté de mettre en œuvre une plateforme aussi intuitive est efficace)
- CONCEPTION NON ADEQUATE !!
- Difficulté de réaliser des Algorithmes de résolution
- Formation ASP.Net, ...
- Autres Problèmes et Formations ...
- Etc. ....





# Plan du Projet :

- Afin de réaliser l'application web (notre projet) pour une plateforme WEB (ASP.Net), il faut suivre des étapes de productions suivantes :

## Etapes :

- Etude préliminaire et Etude de l'existant
- Étude fonctionnelle :
  - Consiste à détailler les modules du projet et leurs fonctionnalités (**CC**).
- Étude technique :
  - Détailler les interfaces de la solution et la réalisation des maquettes.
  - Etude des moyens qui seront déployés dans le projet (**ASP.Net**, ...).
- Conception :
  - Concevoir les modèles et les diagrammes du système d'information.
- Génération et implantation de la base de données.
- Développement des interfaces :
  - Codage et mise en place des interfaces du site.
- Intégration des interfaces, Optimisation & Déploiement
- Mise en Production et Maintenances



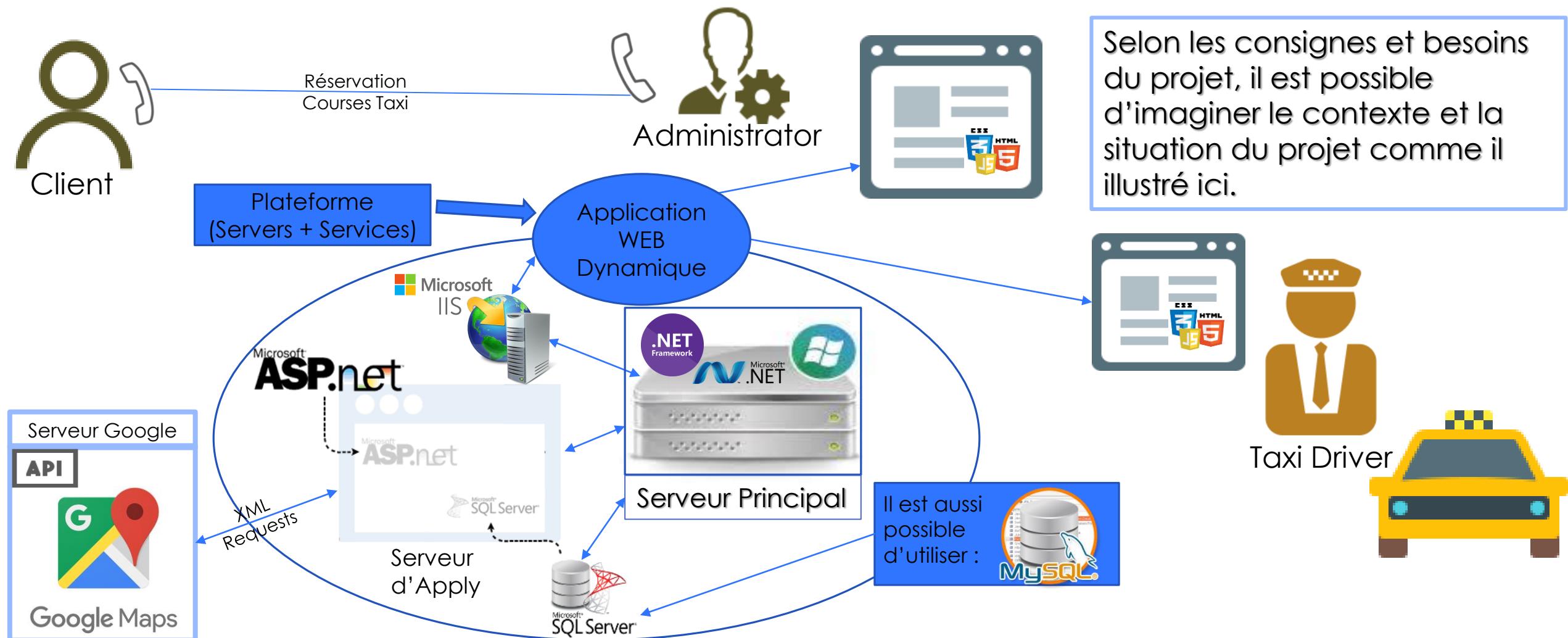


# ETUDES CONTEXTUELLES DU PROJET

- Etude préliminaire et Etude de l'existant

# CONTEXTE DU PROJET :

SITUATIONS ET PENTAGONISTES :



- Etude Fonctionnelle

# ETUDE FONCTIONNELLE :

Les Modules du Système et service Web de la plateforme :

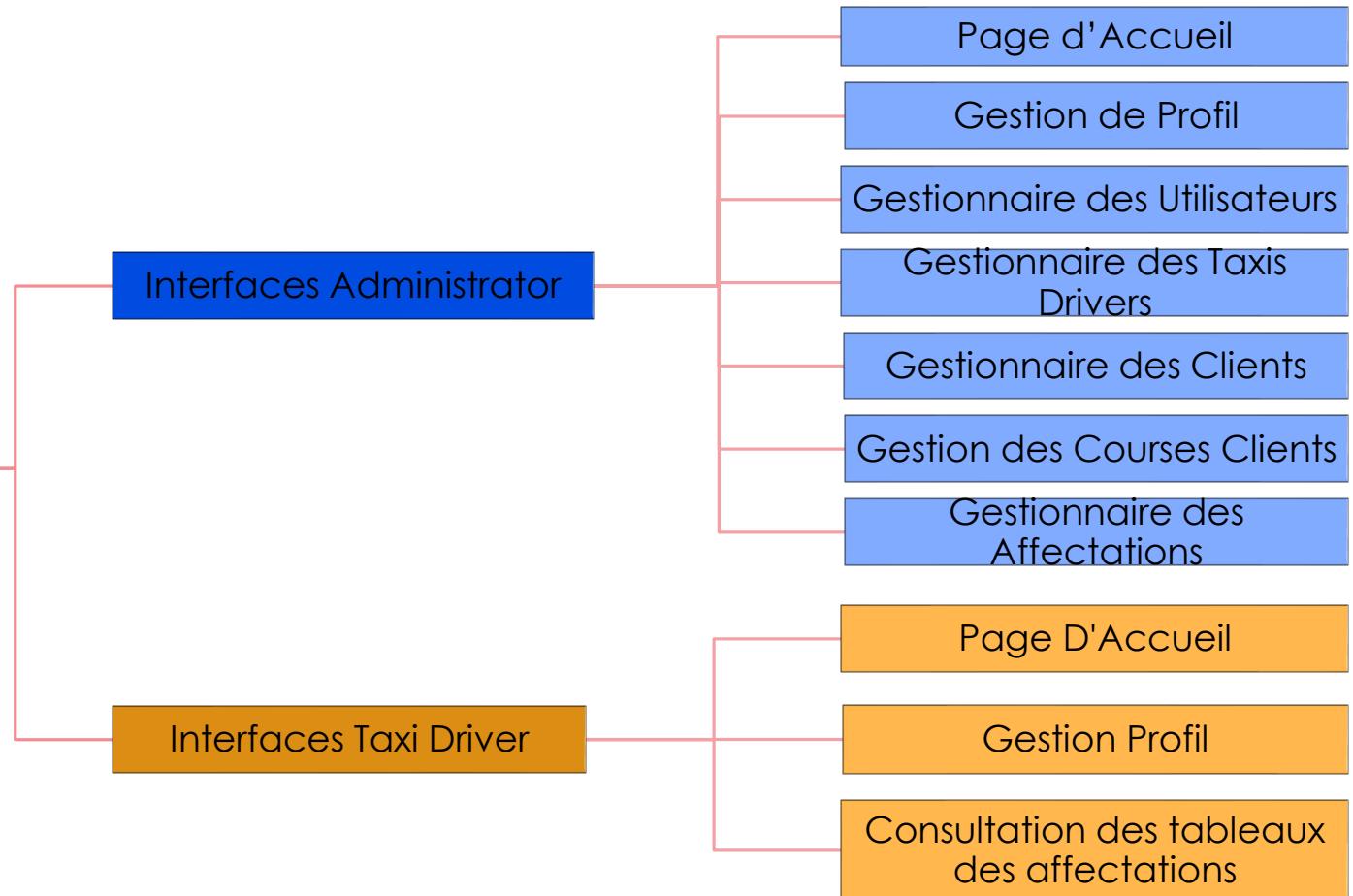
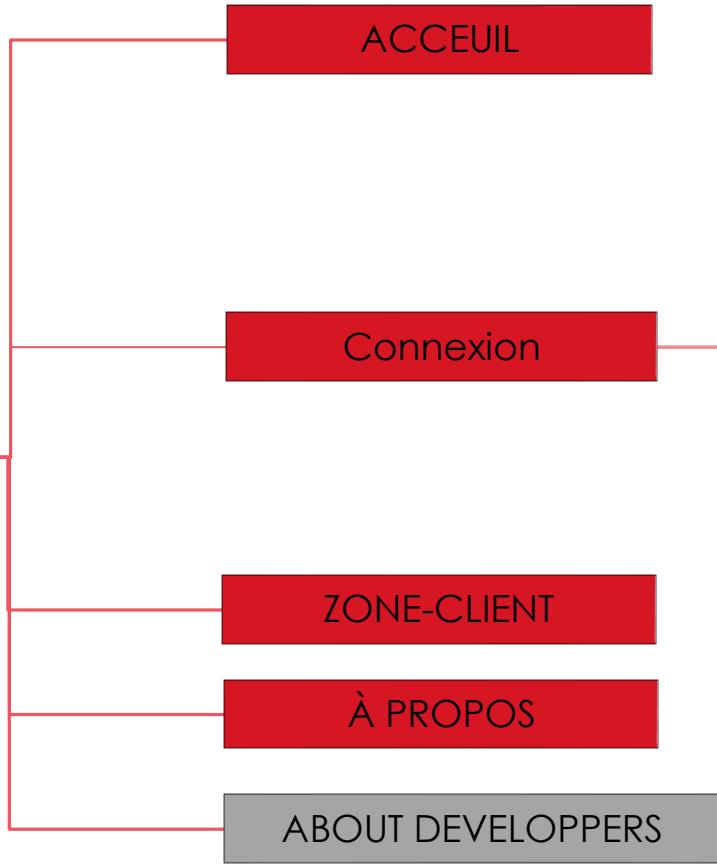
- Administration et Gestion des Utilisateurs (Interfaces Administrators)
- Modules Taxis et Chauffeurs de Taxis (Interfaces Taxis Drivers)
- Module Clients
- Module Courses de Taxis (Inclus dans les interfaces Administrators)
- Module des communications Server :
  - Serveur de Données (MySQL ou MS SQL Server), API Google Maps, ...



# ETUDES TECHNIQUES DU PROJET

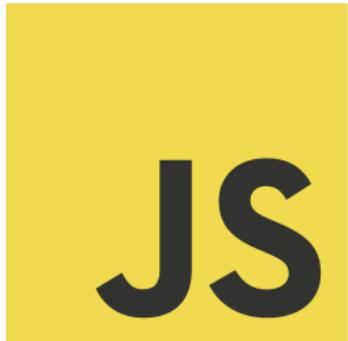
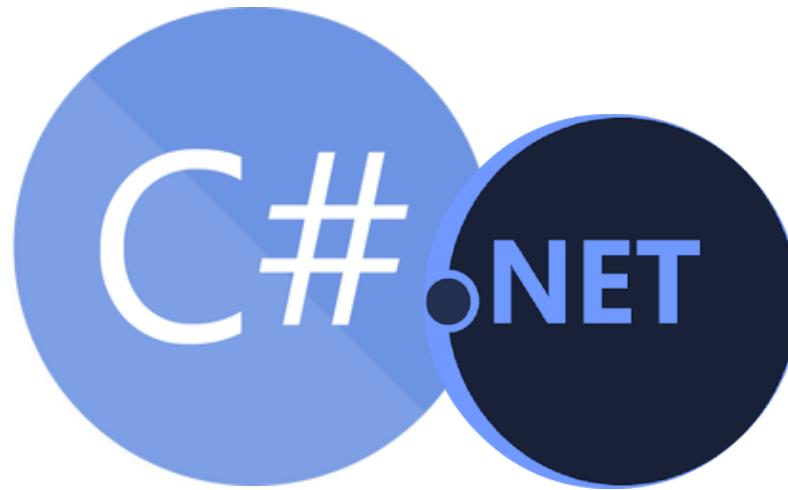
# ARBORECENCE DE LA PLATEFORME :

Plateforme de Gestion des Taxis





# Les Outils et Technologies à Utiliser En Générale:



# Les Outils et Technologies liés à l'ASP.Net (1) :



# Les Outils et Technologies liés à l'ASP.Net (2) :

Microsoft ASP.NET





# Les Outils et Technologies de Développement et de Déploiement :

Microsoft  
ASP.NET



Microsoft  
.net

Microsoft  
ADO .Net

Microsoft  
Visual Studio

Microsoft  
SQL Server

Microsoft  
.NET  
Entity  
Framework



Visual Studio  
MicroSoft VisualStudio 2017 (Entreprise Ed.)



Microsoft®  
2017  
SQL Server®

MicroSoft SQL Server 2017 (Developper Ed.)



Microsoft SQL Server Management  
Studio



Microsoft  
IIS

(MicroSoft IIS 10 sur Microsoft Windows 10)

# ETUDES DÉTAILLÉES DES TECHNOLOGIES ET OUTILS (1) :



ASP.NET apporte à la fois une grande productivité pour le développeur et améliore les performances, la fiabilité et simplifie le déploiement des applications Web. Et il offre aussi :

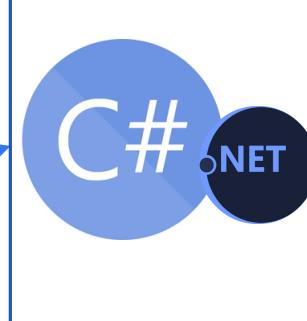
- **Modèles de programmation simplifié.**
- **Choix des langages ouvert (Par exemple C#.Net).**
- **Environnement graphique puissant.**
- **Framework de classes étendu.**
- **Système de gestion de contenu.**
- **Communication avec les bases de Données.**
- **Etc**

Il permet aussi de générer des pages à la demande, et il cohabite avec les standard du WEB et les services WEB (tels que les API Google).



Google Maps for Work

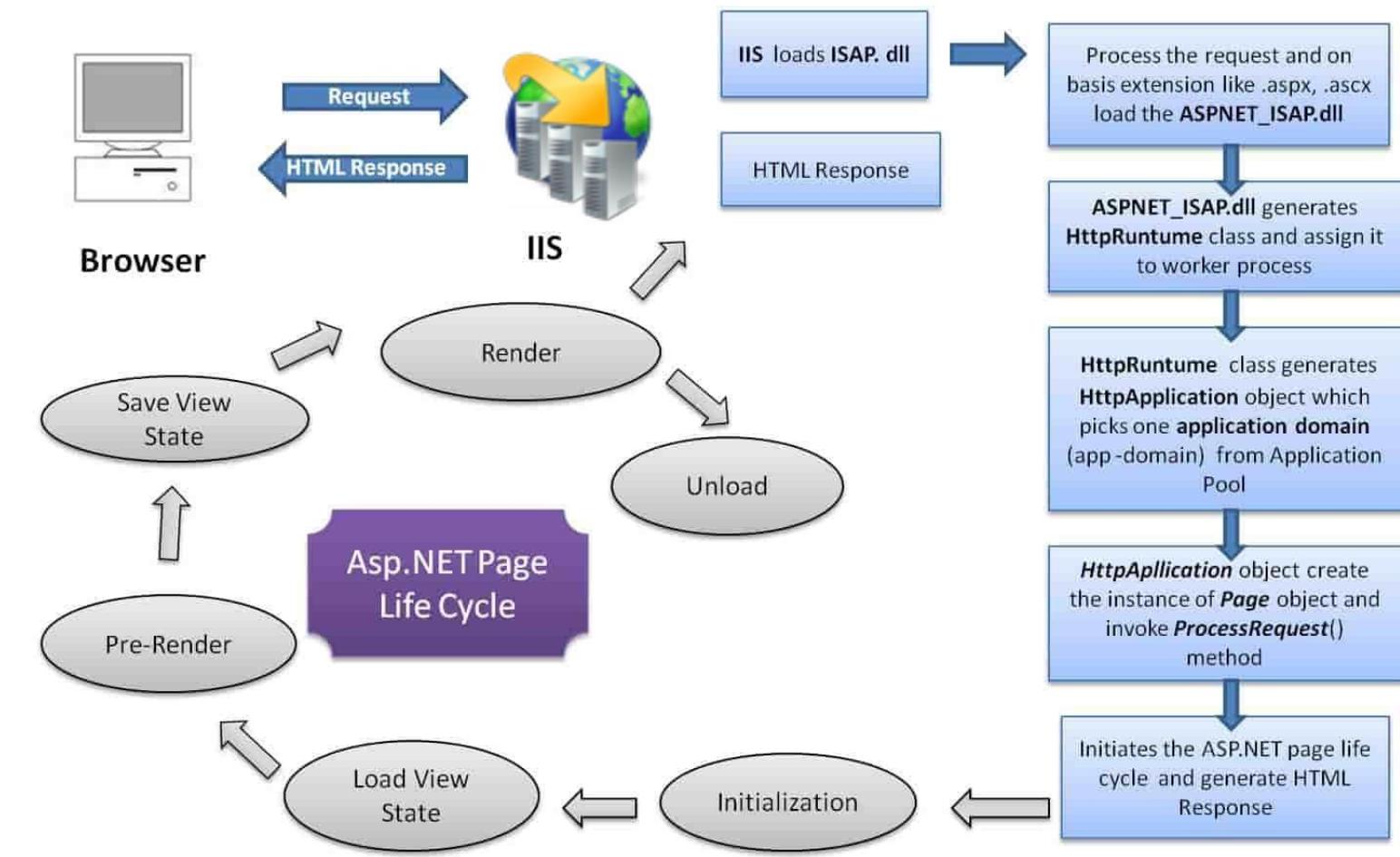
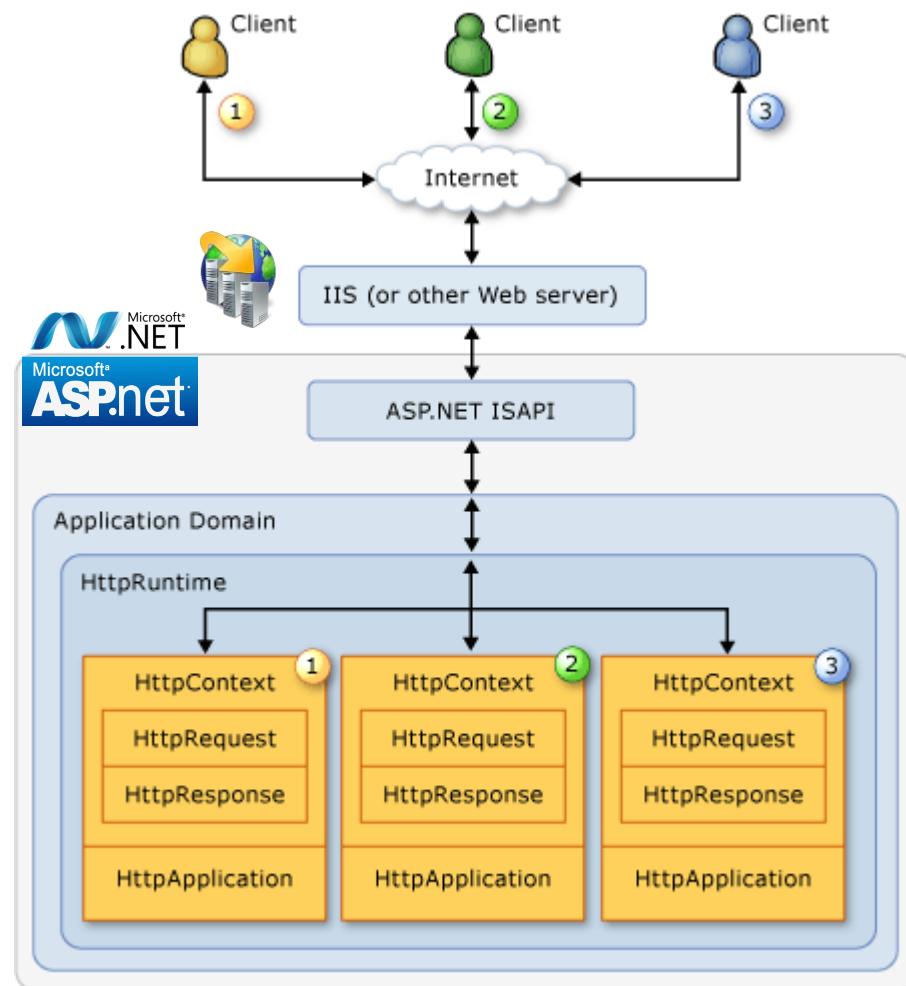
Pour Estimer la durée du trajet et la distance en fonction d'un itinéraire recommandé.

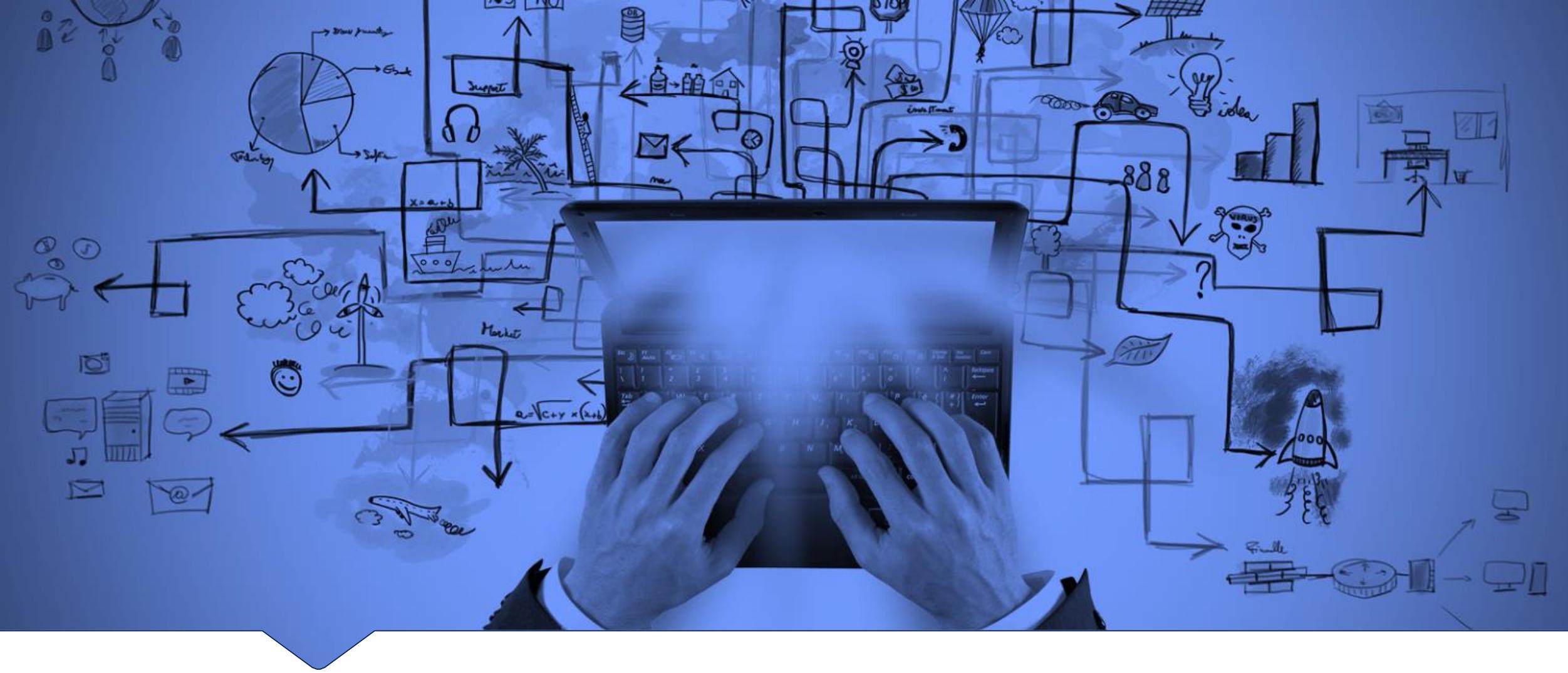


C#.Net est un langage de programmation orientée objet, ressemblant JAVA. Il est utilisé pour développer des applications web, ainsi que des applications de bureau, des services web, ou des bibliothèques de classes... C#.Net est destiné à développer sur la plateforme .NET.



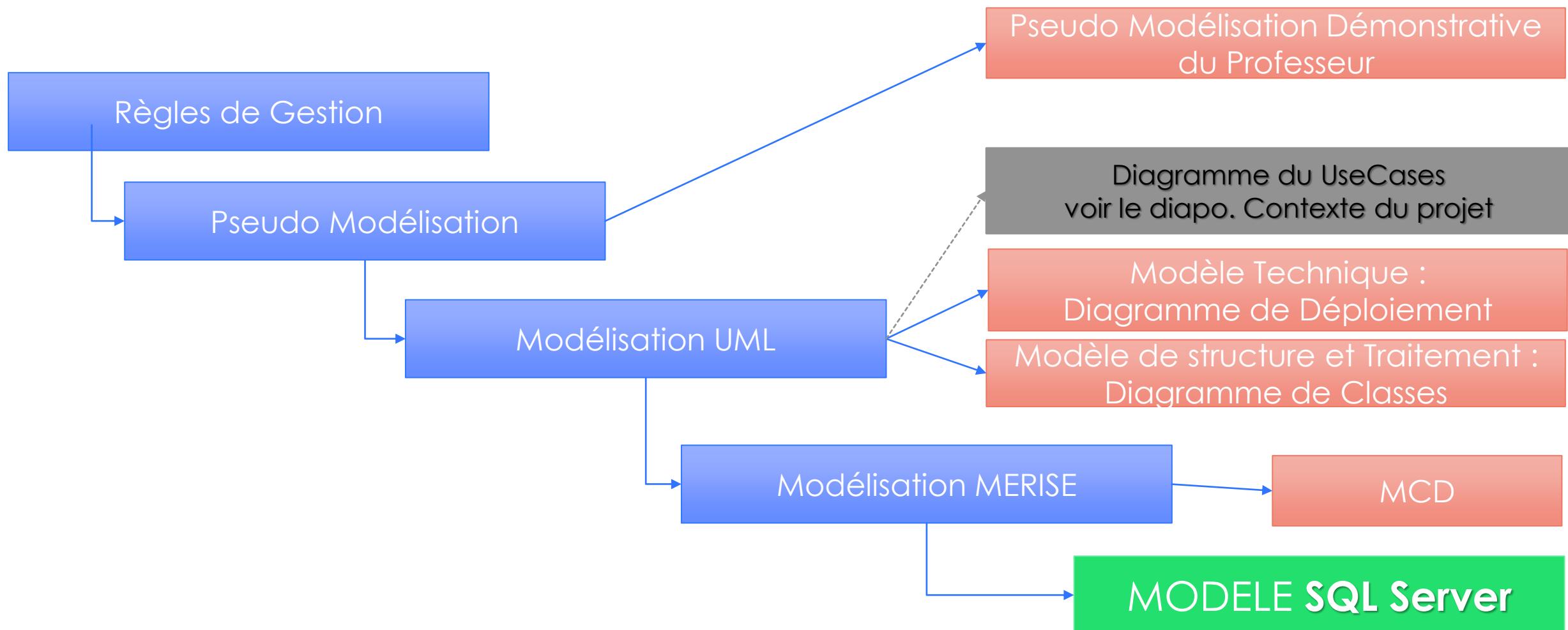
# ETUDES DÉTAILLÉES DES TECHNOLOGIES ET OUTILS (2) :





# CONCEPTION ET MODELISATION

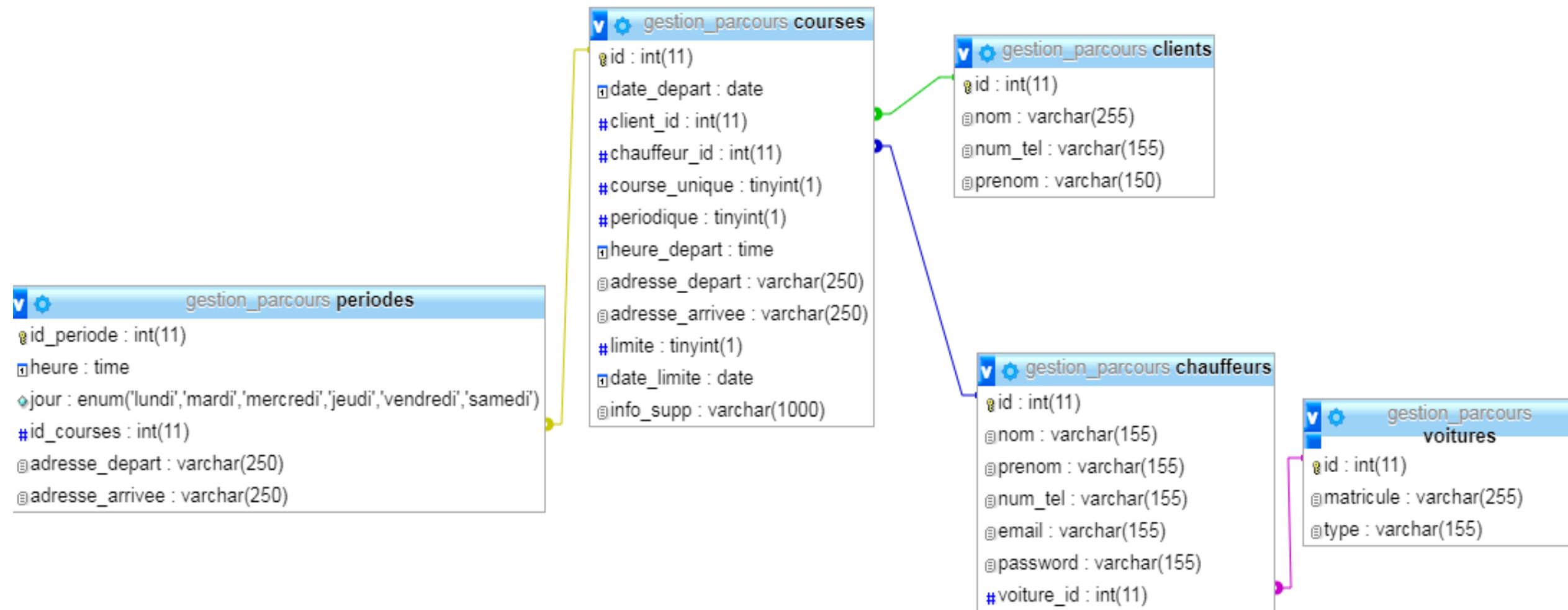
# PLAN DE MODELISATION :



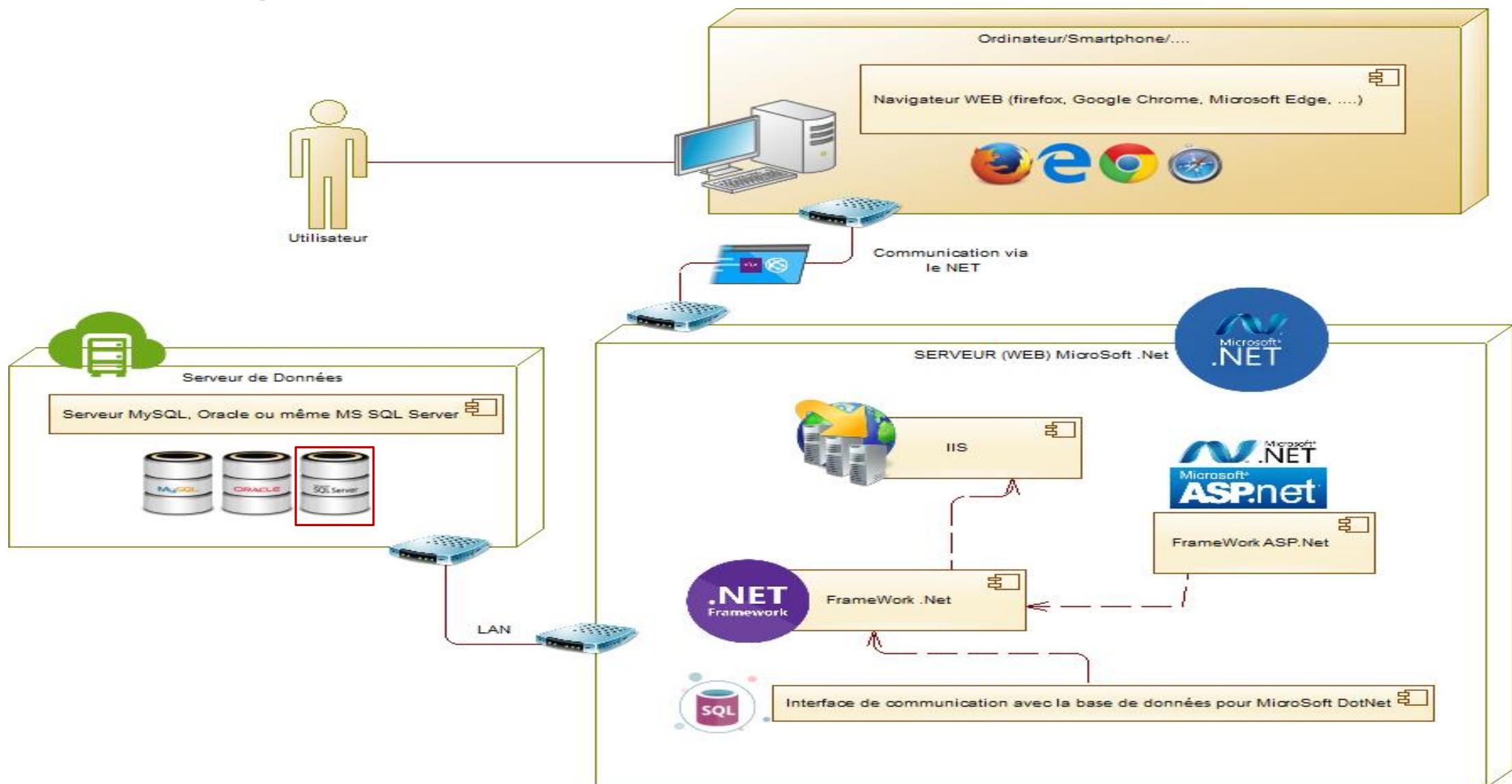
# LES REGLES DE GESTION :

- Une Ste. de transport est constituée des Gestionnaires (Administrators) et les chauffeurs de taxis (Taxi Drivers).
- Un Administrator est défini par : id, nom, prénom, téléphone, email et mot de passe.
- Un Taxi Driver est défini par : id, nom, prénom, téléphone, email, mot de passe et puis un taxi.
- Un Taxi est défini par : id, matricule, type(?). Et les types de voiture (taxi) sont ....
- Un Taxi n'est affecté qu'à 1 et 1 seule Taxi Driver.
- Un Taxi ne peut avoir qu'1 et 1 seule client à un instant t donné.
- Un taxi peut démarrer le travail de 7h du matin jusqu'à 20h du soir.
- Le client appelle pour réserver une course.
- Lors de l'affectation des courses de clients, il faut maximiser l'affectation à un taxi avant de passer à un autre taxi et ainsi de suite. C'est à dire, Pour tout taxi, il faut lui affecter un maximum de courses clients avant de passer au deuxième taxi et ainsi de suite.
- L'affectation des Courses de clients aux taxis devra respecter la succession dans le temps et la succession géographique. C'est-à-dire, pour affecter une course à un taxi, il faut l'heure de départ vient après l'heure d'arrivée de la dernière course du taxi plus la marge du retard tolérée sinon il faut affecter cette course à un autre taxi.
- La marge de retard tolérée de chaque rendez-vous est de 5min.
- Une course est défini par : id, date départ, heure départ, adresse départ, adresse d'arrivée et un client.
- Un Client est défini par : id, nom, prénom et téléphone.
- Certains clients peuvent faire plusieurs courses dans la même semaine et d'une façon périodique (chaque semaine). La périodicité peut être limitée par une date ou non.
- Depuis la course périodique, il est généré une course (normale) à chaque date correspondante.
- Une course périodique est défini par : id, jour, heure départ, adresse départ, adresse d'arrivée, date limite et un client.
- Les informations des clients sont supposées enregistrées d'avance.
- Les chauffeurs peuvent consulter en tout moment leurs tableaux des affectations. Chacun d'eux ne peut consulter que son propre affectation des courses.
- Un administrateur peut ajouter, éditer et consulter une course, un client, chauffeur et Taxi.
- Un administrateur peut consulter toutes les affectations.

# VISION ET MODELISATION DU PROFESSEUR :

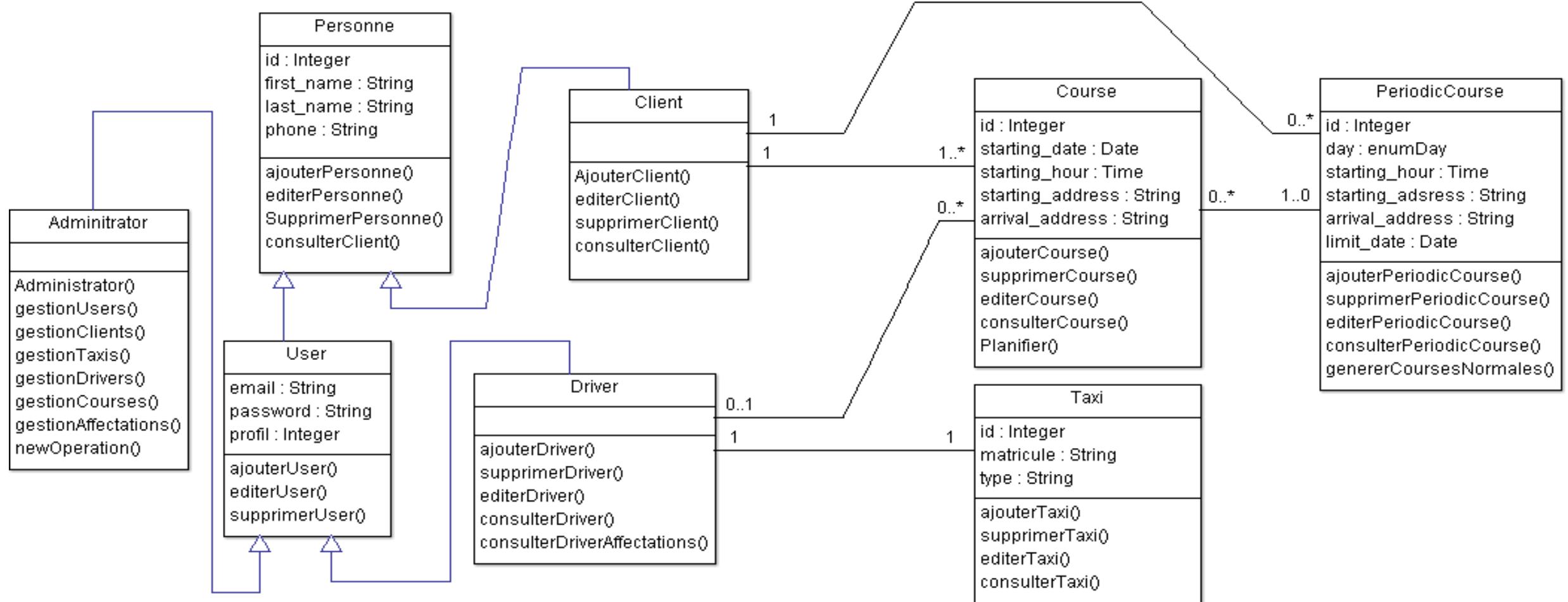


# DIAGRAMME DE DÉPLOIEMENT :

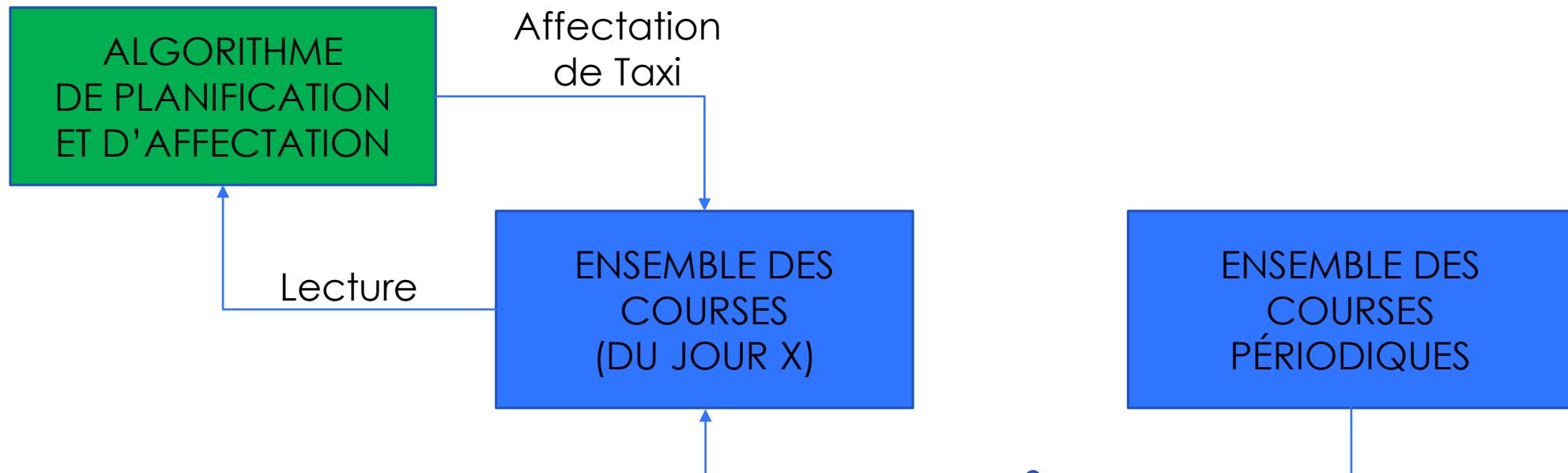


# DIAGRAMME DE CLASSES :

- UML – Diagramme de Classe (Prévisionnel) :

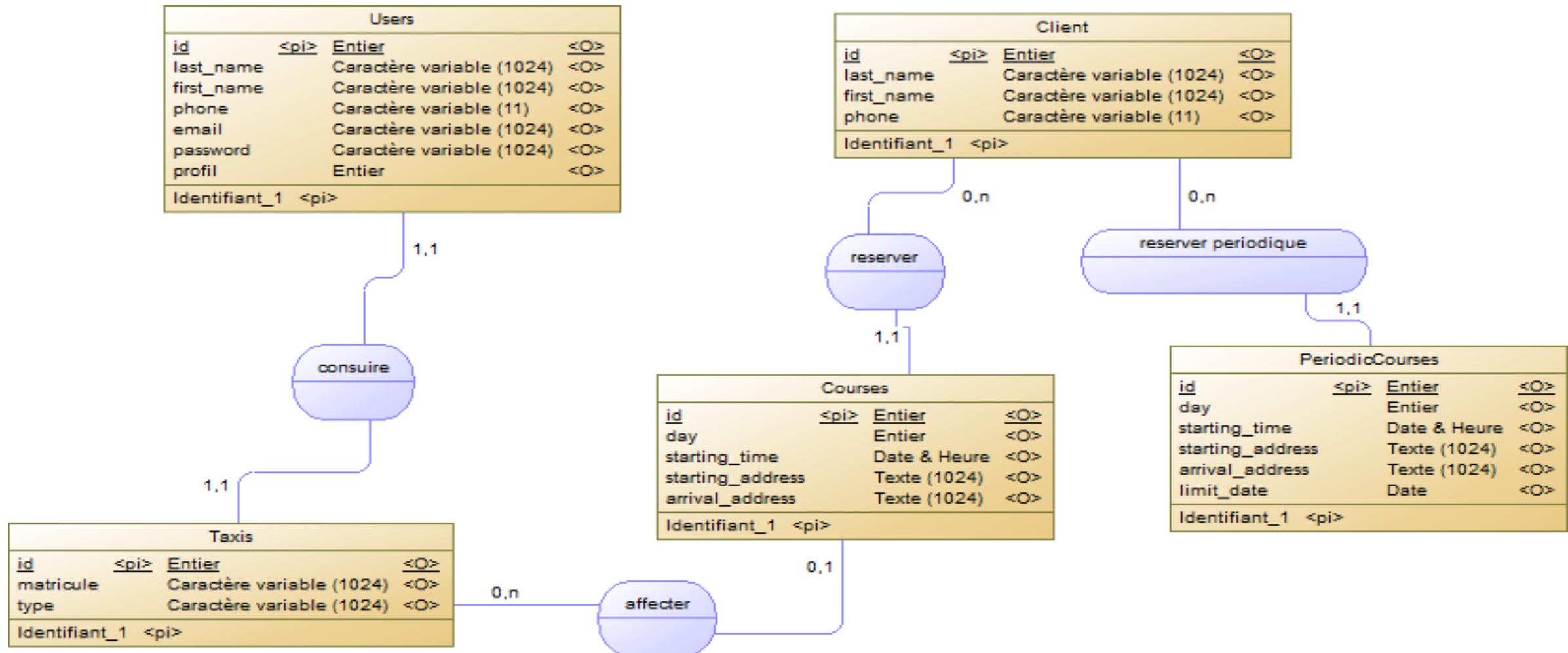


# CONCEPTS ET ALGORITHME :



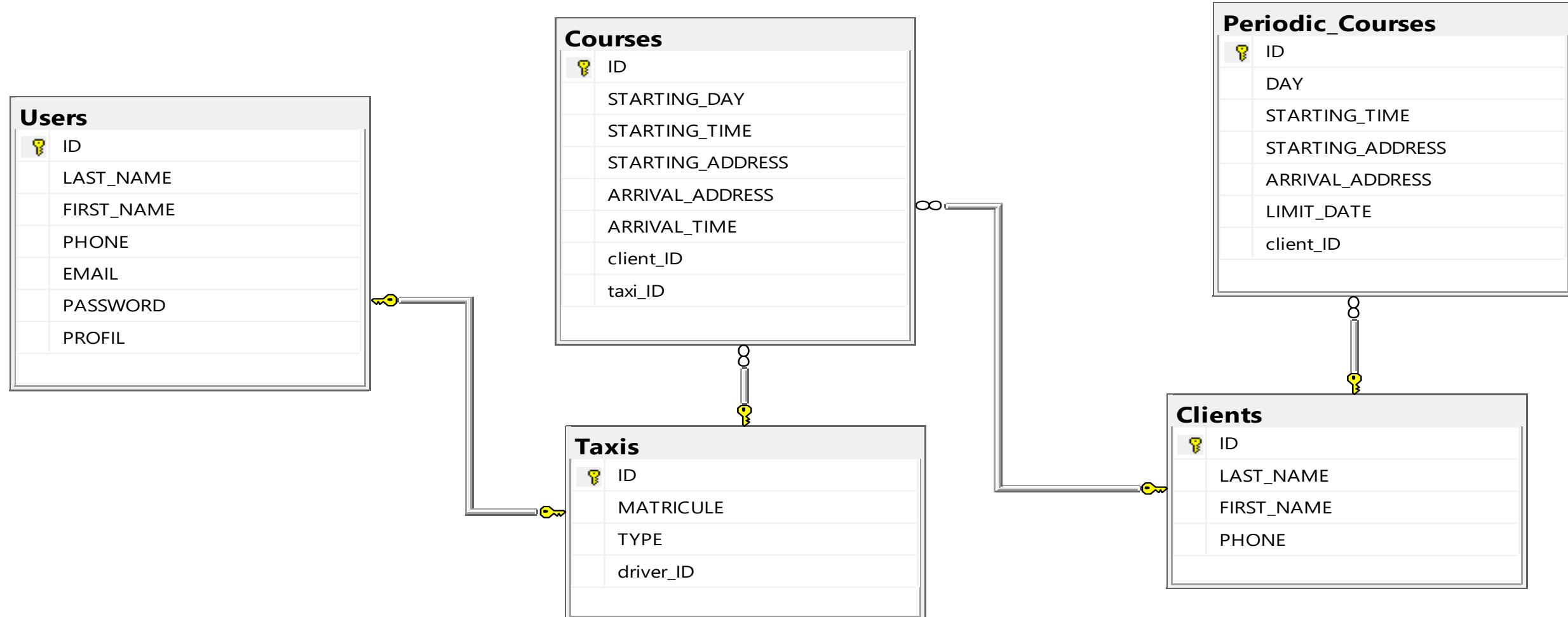
Ceci est pensé ainsi pour que l'algorithme gère les courses de la même manière. Et donc ça permet d'alléger l'algorithme et la procédure de planification, puisque l'algorithme n'a pas géré différemment les courses périodiques. Sinon il aura qu'un seul taxi pour une courses périodique et cela pour toute les dates.

# DIAGRAMME MCD :



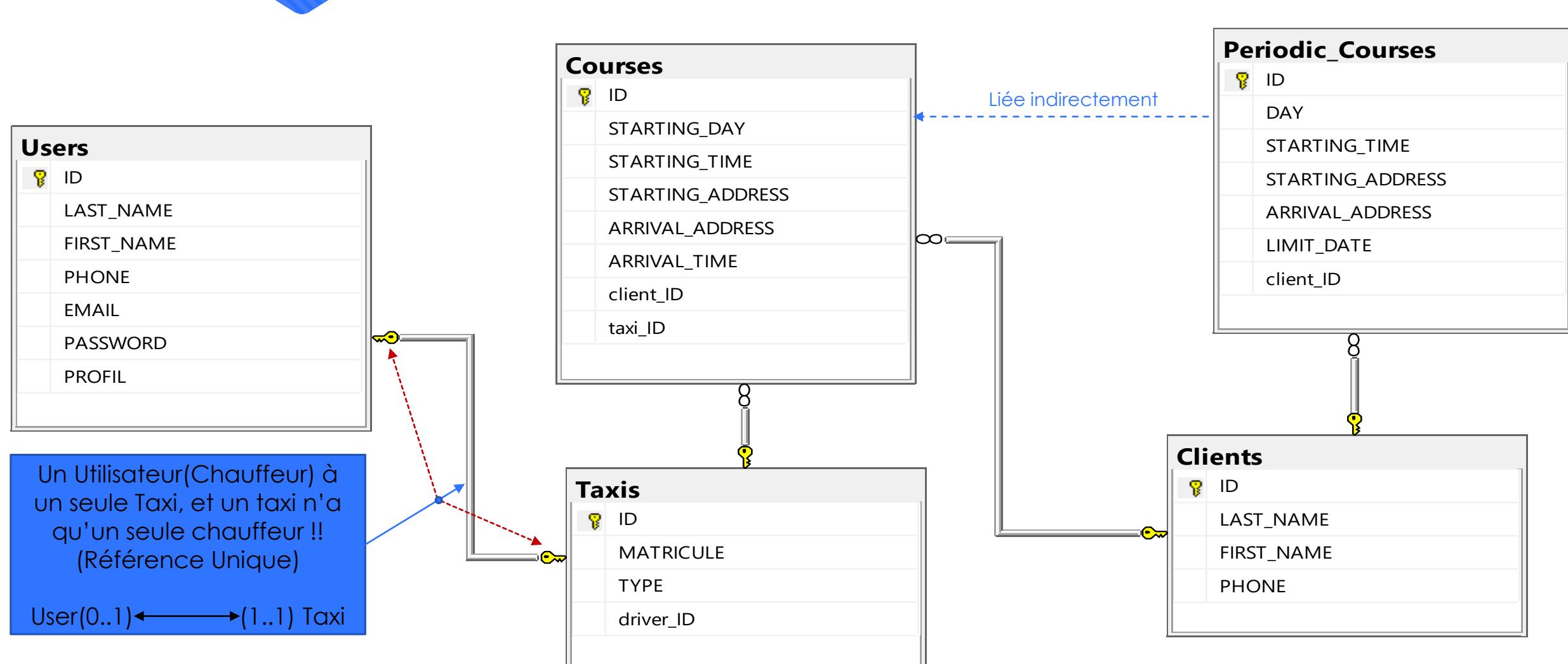
# MODELE MICROSOFT SQL SERVER (1) :

Diagramme ERD SQL SERVER – SSMS (1) :



# MODELE MICROSOFT SQL SERVER (2) :

Diagramme ERD SQL SERVER – SSMS (2) :





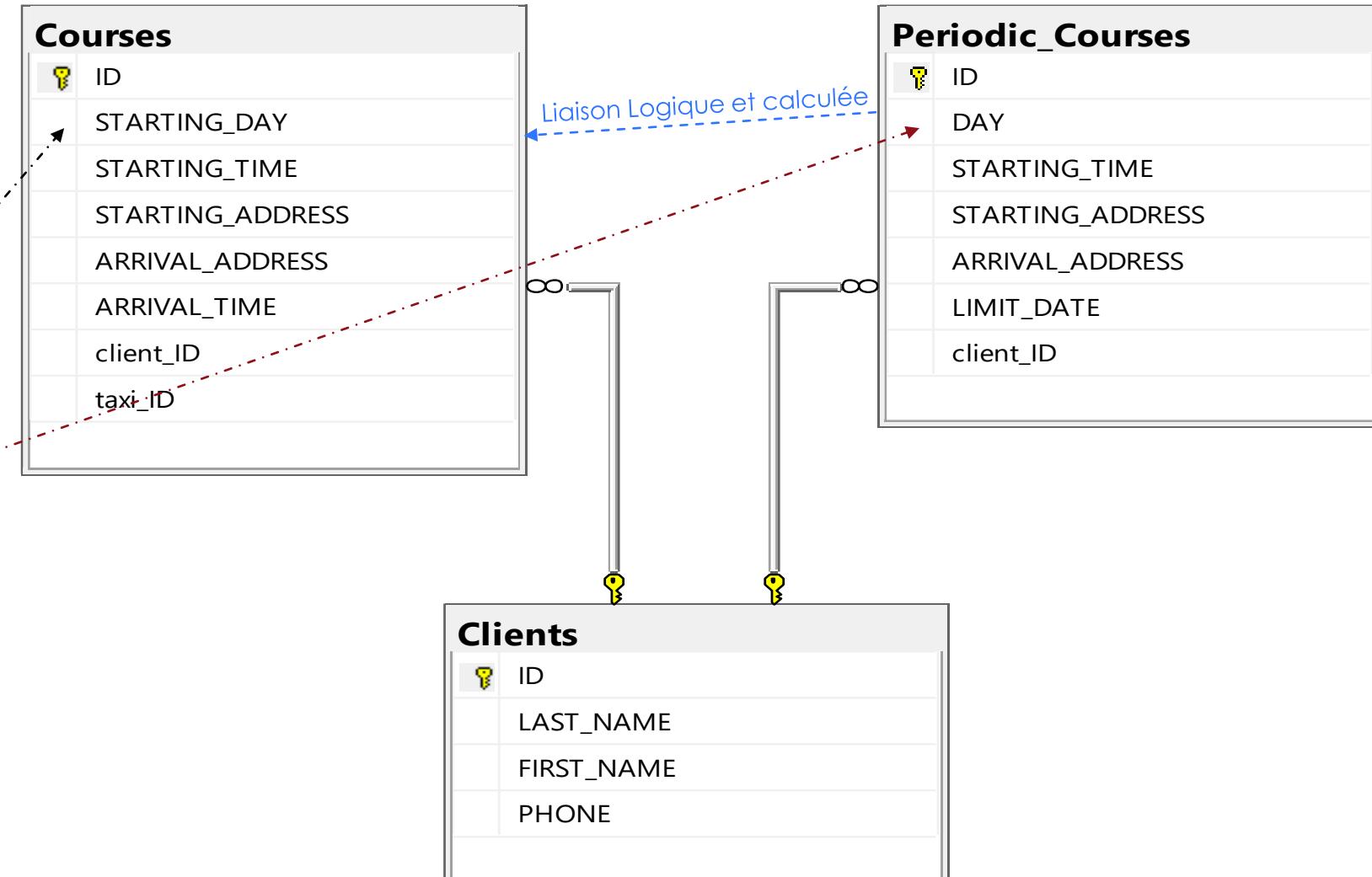
# MODELE MICROSOFT SQL SERVER (3) :

Diagramme ERD SQL SERVER – SSMS (3) :

Pour Savoir les courses Générées  
à partir d'une course  
périodique

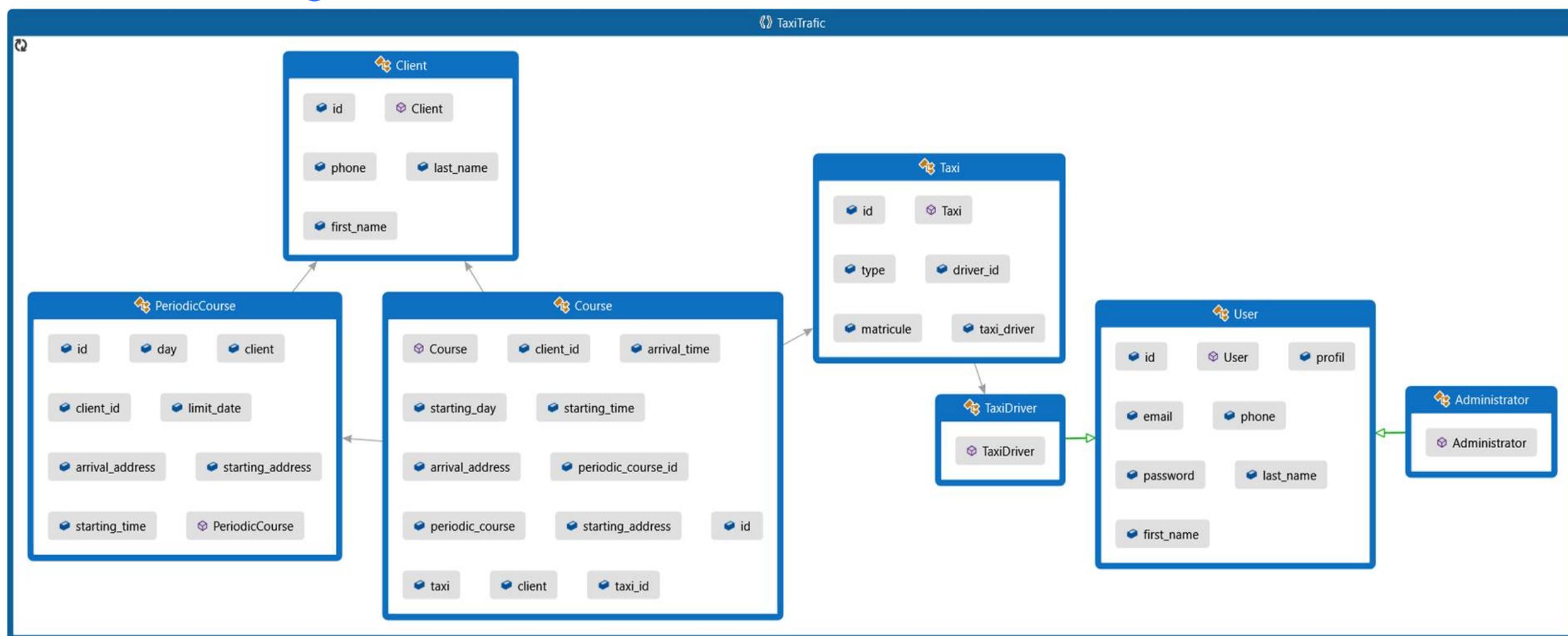


Si course **C1** d'un client dont le jour de la semaine de son **C1.STARTING\_DAY** correspond à DAY (**CP1.DAY**) de la course périodique **CP1** du même client, alors **C1** a été généré à partir de **CP1**, et donc sans le formalisme des clés étrangères, il est possible de savoir les enregistrements mis en relation entre les 2 tables.

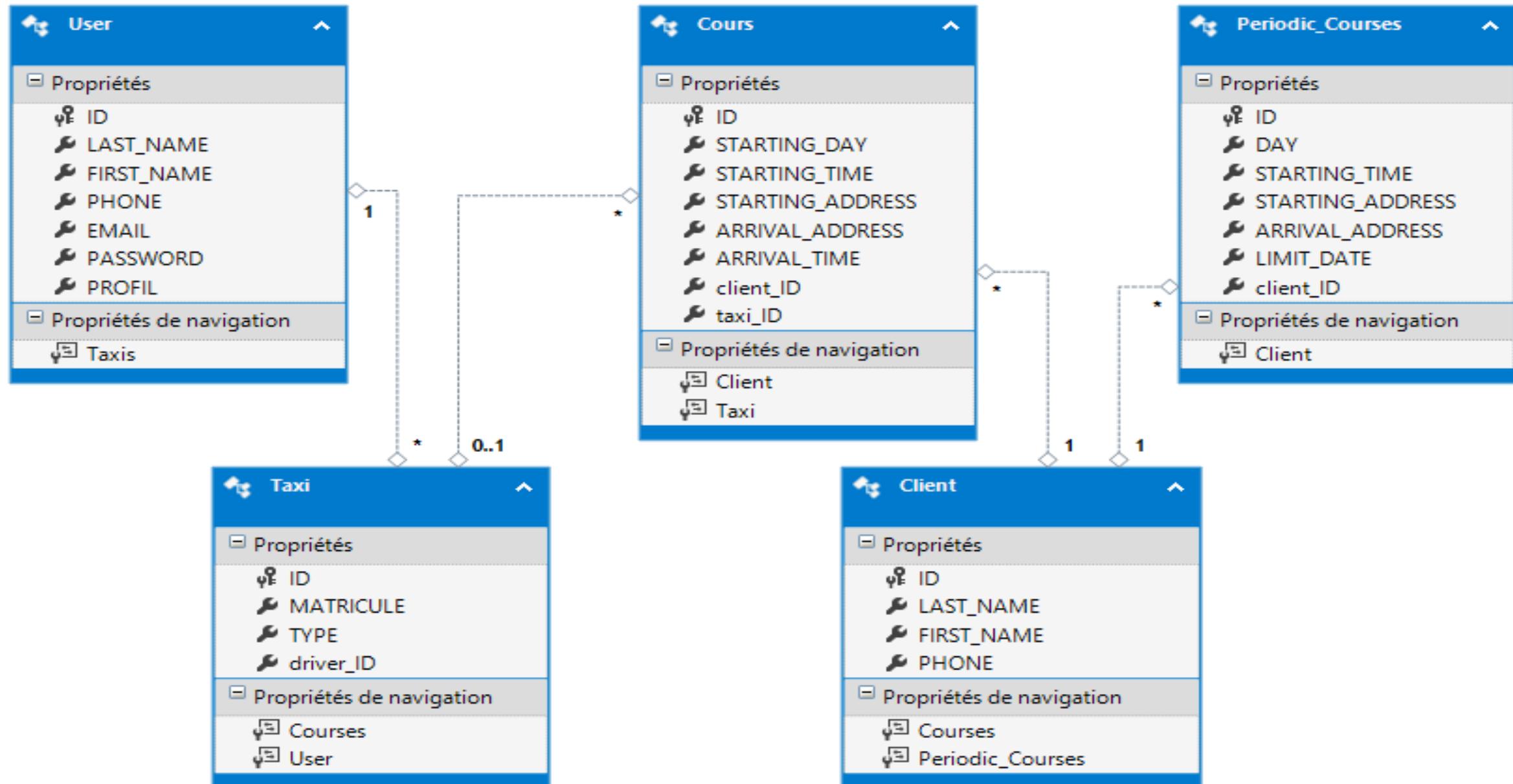


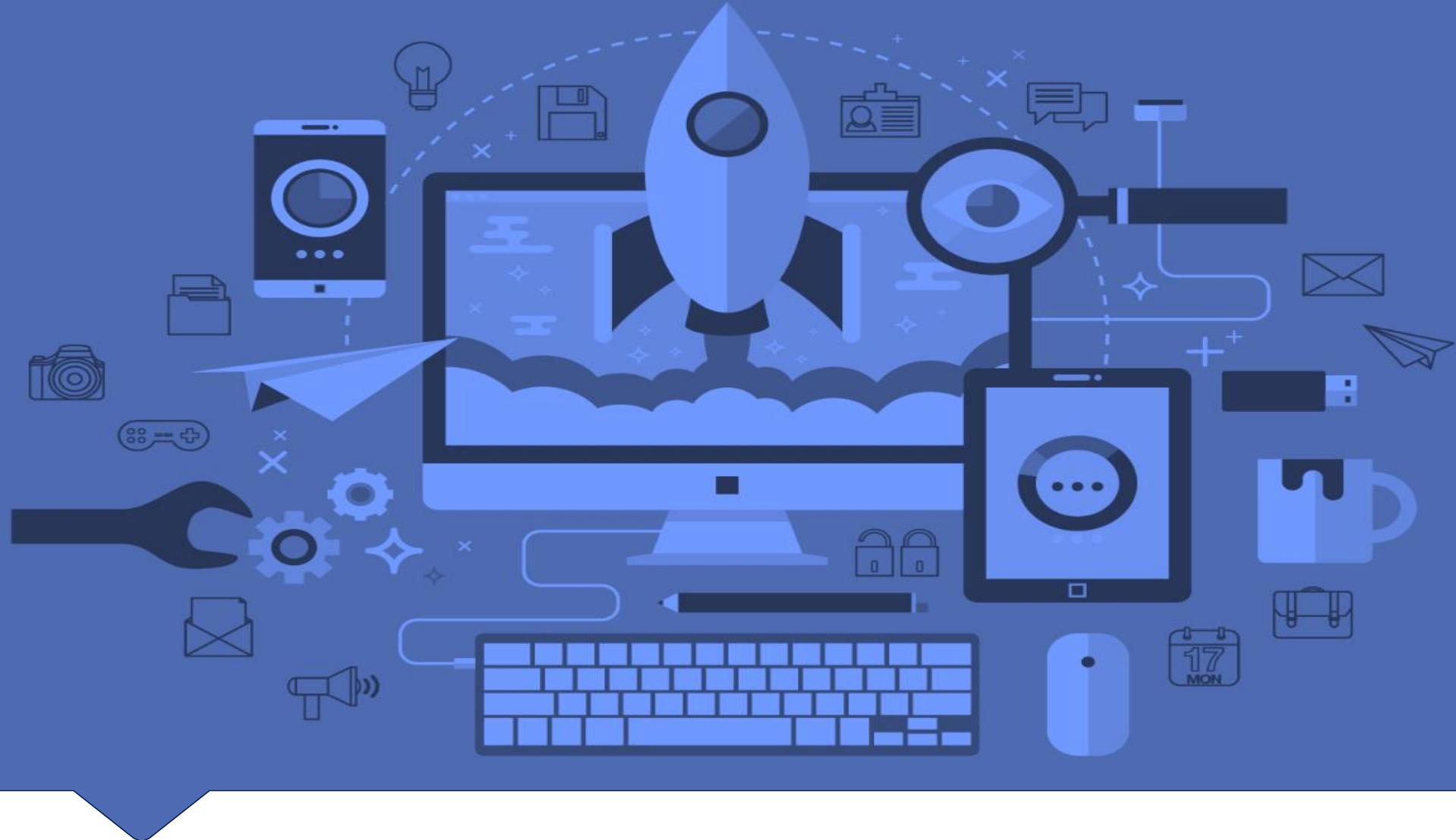
# MICROSOFT DESIGN CODE MAP (C#.NET & ASP.NET) :

VISUAL STUDIO DESIGN CODE MAP (Prévisionnel) :



# MS VISUALSTUDIO ENTITY DESIGNER DIAGRAM :

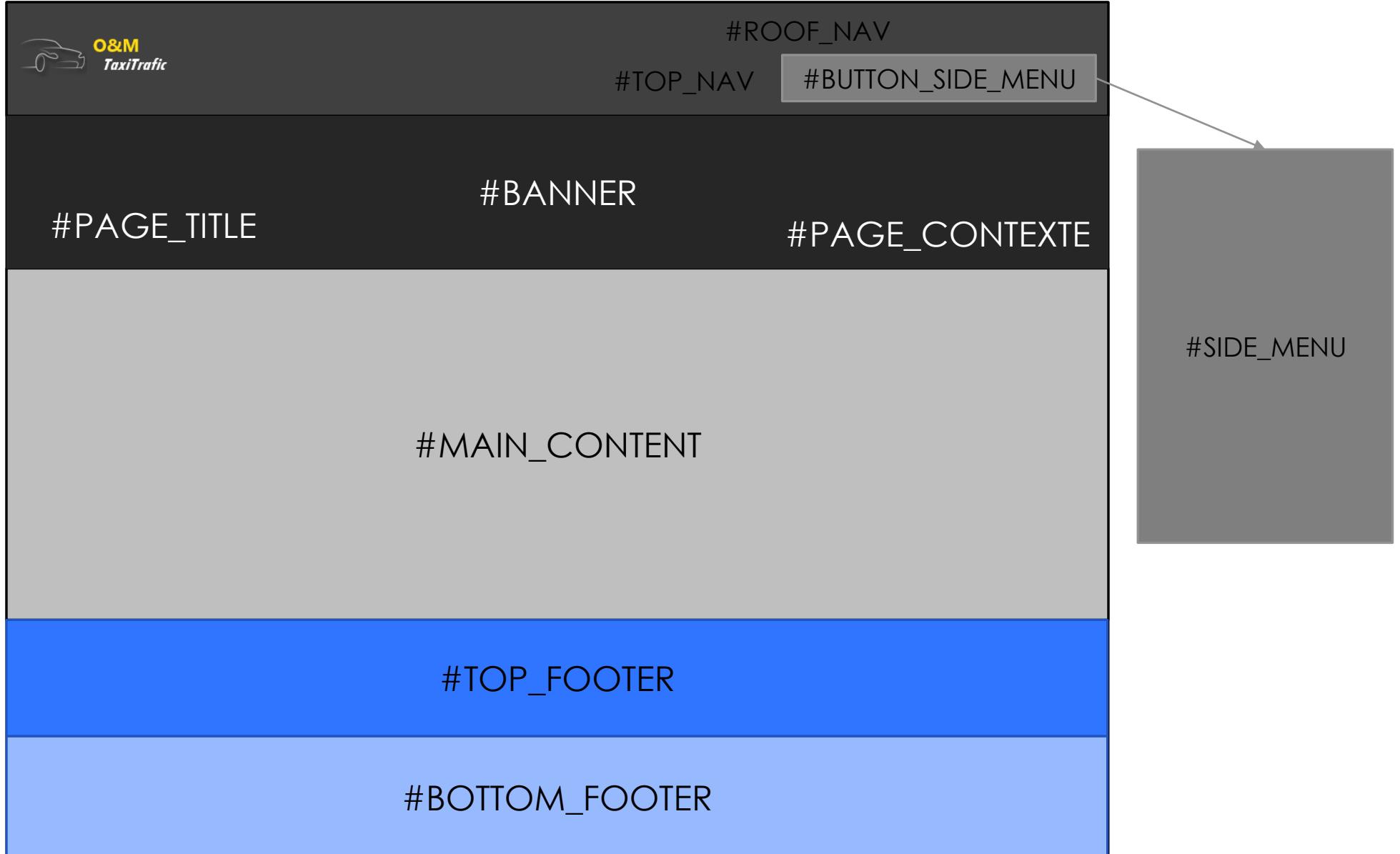




# ARCHITECTURE & ASP.NET WEB-APPLY DESIGN

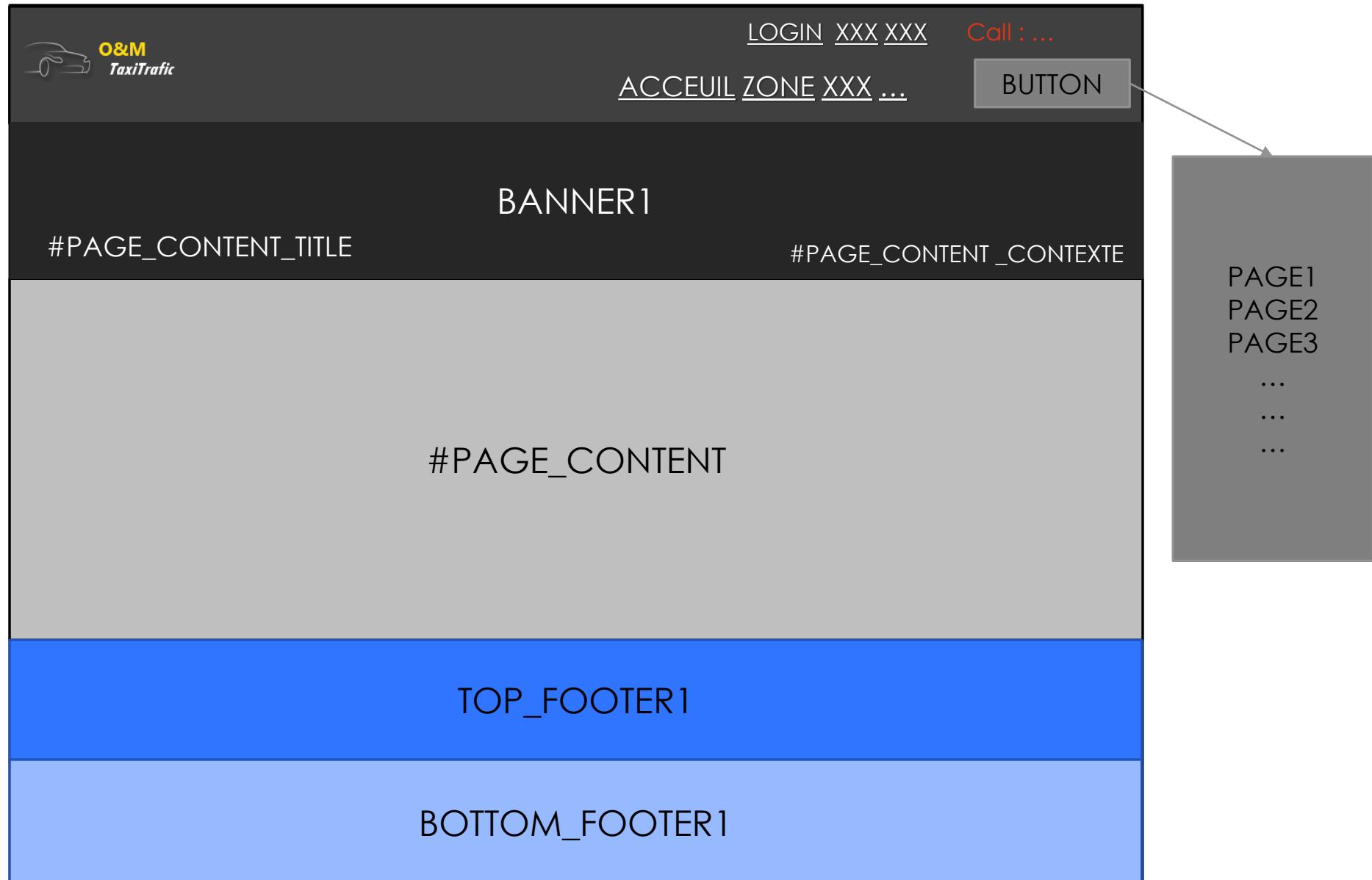
# PAGE MASTER :

RmQ : # ⇄ ContentPlaceHolder



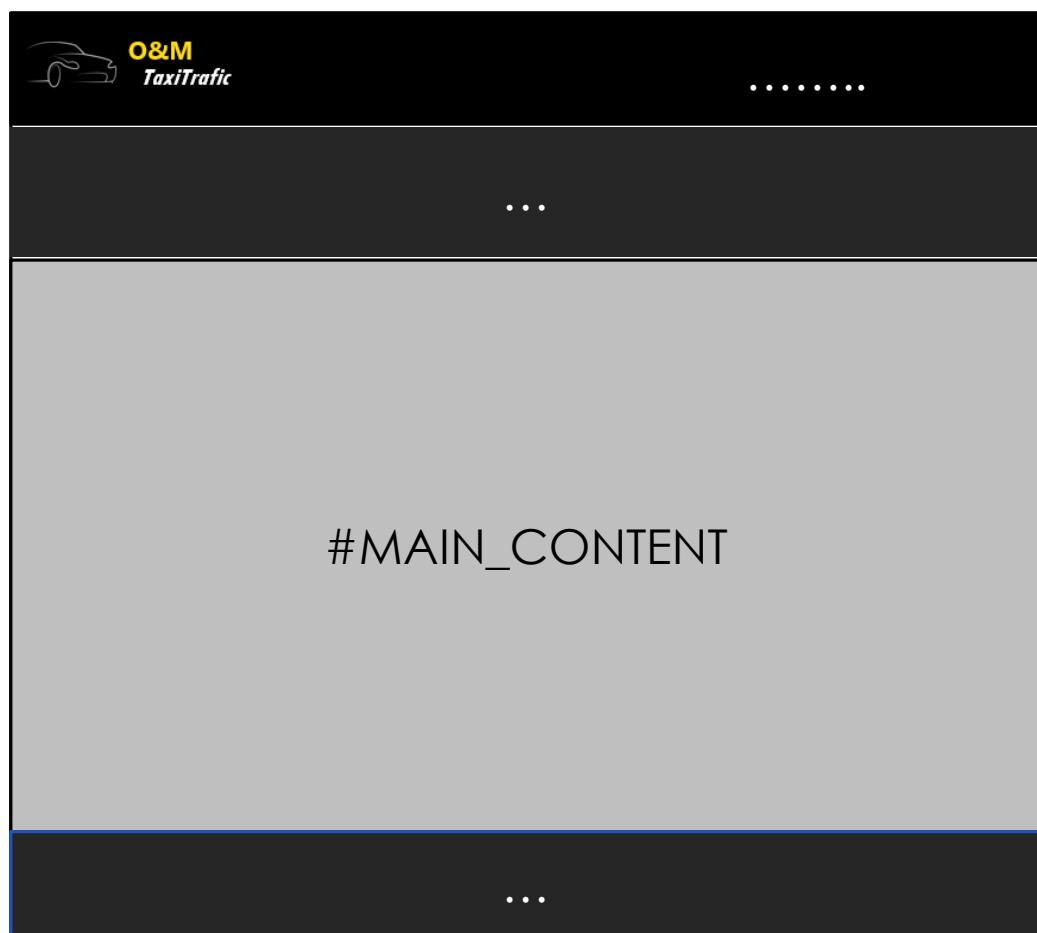
# PAGE MASTER IMBRIQUÉ :

RmQ : # ⇔ ContentPlaceHolder



# SPECIALISATION (1) :

RmQ : # ⇔ ContentPlaceHolder



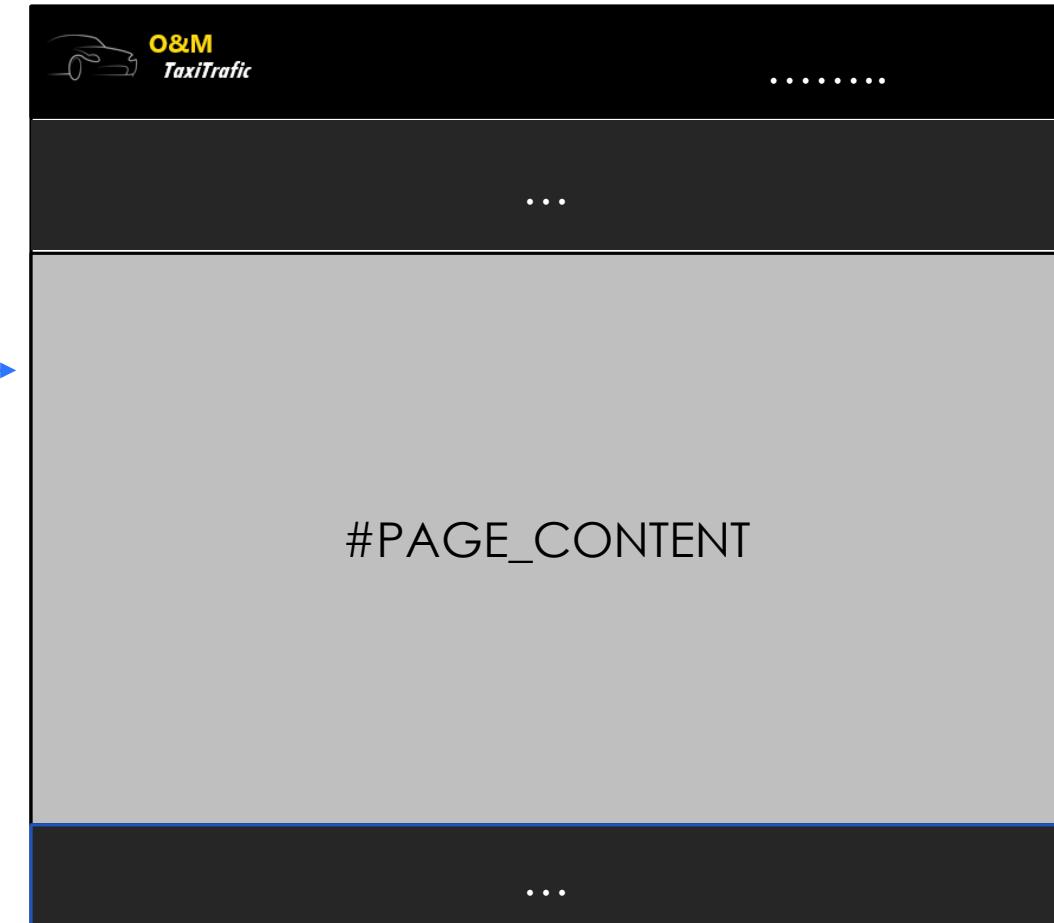
MasterPage

## Spécialisation

Spécialisation Selon  
le profile de  
l'utilisateur :

- Non Connecté
- Administrator
- TaxiDriver

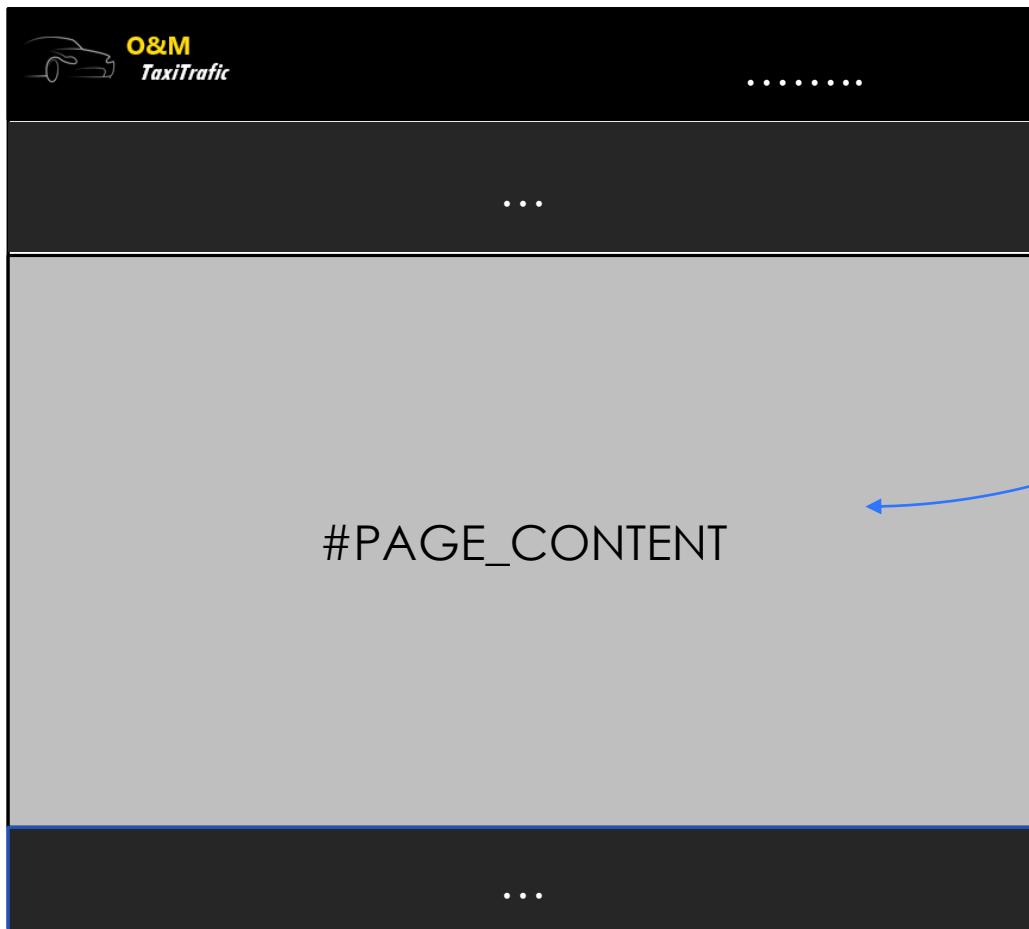
Chaque mode à  
une page Master  
Imbriquée, car le  
menu et quelques  
autres contenus  
diffères.



MasterPageImbriquée

# SPECIALISATION (2) :

RmQ : # ⇔ ContentPlaceHolder

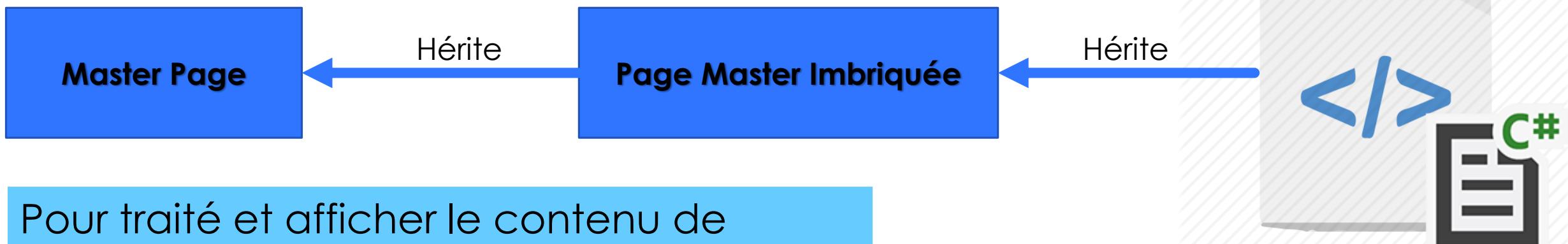


MasterPageEmbrayée



Contenu à afficher et à traiter, il hérite de MasterPageEmbrayée  
Ainsi de MasterPage .

# SPECIALISATION (3) :



Pour traiter et afficher le contenu de PageContent, il sera traité avant les éléments de la page mère qui est Page Master Imbriquée après le traitement les éléments de la page dont hérite cette dernière qui est MasterPage .

# Les Ressources à consulter :

- [www.google.com](http://www.google.com)
- [www.youtube.com](http://www.youtube.com)
- <https://www.stackoverflow.com>
- <https://msdn.microsoft.com/fr-fr/>
- <https://www.asp.net/>
- <https://docs.microsoft.com/fr-fr/aspnet/overview>
- <https://docs.microsoft.com/fr-fr/aspnet/#pivot=aspnet>
- <https://developers.google.com/>
- <https://www.tutorialspoint.com/asp.net/index.htm>
- <https://developers.google.com/maps/?hl=fr>
- Etc. ...



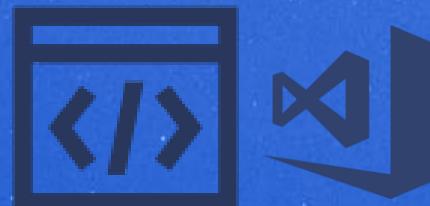


# Google

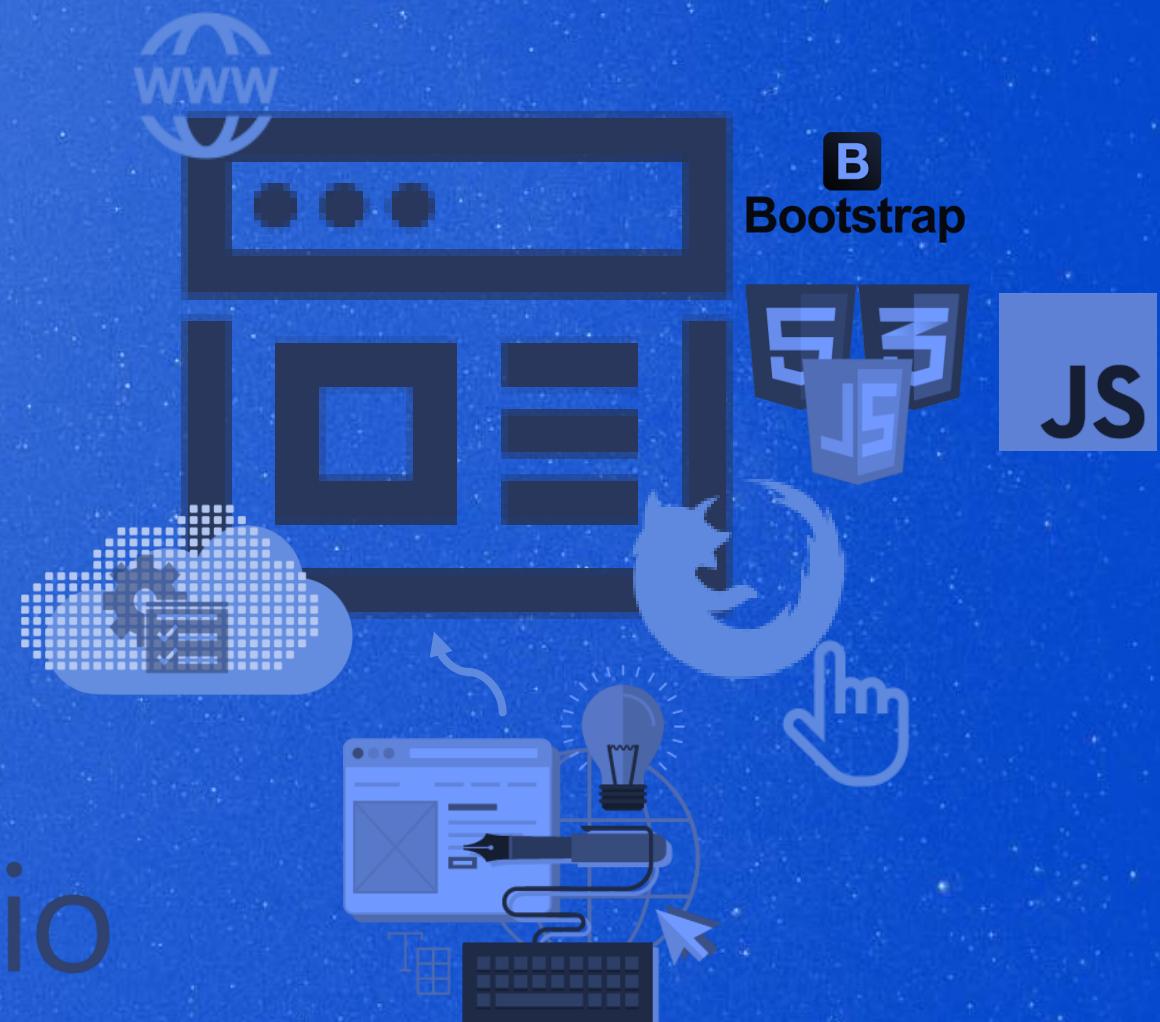
Microsoft

# ASP.net™

Microsoft  
C#.net



# Visual Studio



ECOLE MAROCAINE DES  
SCIENCES DE L'INGENIEUR  
Membre de  
HONORIS UNITED UNIVERSITIES

## EMSI Project - Développement WEB