

## S.I. Gestion des transports

### Description du projet

L'application de gestion des transports permet à un collaborateur de :

- réserver un véhicule de service pour un déplacement professionnel. `BookVehicle()`
- d'organiser un covoiturage entre une adresse de départ et une adresse d'arrivée, `CreateNewCarpooling()`
- de réserver une place sur un covoiturage existant. `BookCarpoolingReservationSeat()`

L'administrateur peut gérer le parc de véhicules de service :

- Ajouter un véhicule `AjouterVehicule()`
- Mettre un véhicule en travaux `MettreVehiculeEnTravaux()`
- Supprimer un véhicule `SupprimerVehicule()`

### Contraintes d'accès

Cette plateforme est une application web accessible via un navigateur.

### Contraintes techniques

Utiliser uniquement des technologies du stack technique .Net

### Fonctionnalités application



#### Authentification

Après authentification, le menu s'adapte au profil. `Login()` `Logout()`



#### Visualisation des réservations de type covoiturage

`GetAllCarpoolingReservations()`

En tant que collaborateur, je souhaite pouvoir accéder à la liste de mes réservations de type covoiturage afin de les visualiser. Le collaborateur n'est pas l'organisateur de ces covoiturages, il est passager.

Je peux visualiser les réservations en cours et l'historique des réservations.

Un covoiturage est constitué de :

- Une date heure de départ
- Une adresse de départ
- Une adresse d'arrivée
- Un véhicule de service
- Un nombre de places restantes

Je peux effectuer les actions suivantes:

- Cliquer sur le détail d'une réservation (en cours ou historique) ce qui permet d'afficher les informations complémentaires suivantes :  
`GetCarpoolingReservationDetails()`

- Véhicule (marque, modèle, nb de places d'origine, immatriculation) => voir pour rattacher le véhicule perso à son profil afin de ne pas ressaisir les mêmes infos à chaque fois.
  - Organisateur
  - Liste des autres passagers
- Annuler une réservation de covoiturage, ce qui me sort de la liste des passagers et libère une place sur le covoiturage. `CancelCarpoolingReservation()`

## Réserver sur un covoiturage

En tant que collaborateur, je souhaite réserver une place dans un covoiturage pour effectuer un déplacement. BookCarpoolingReservationSeat()

Des critères de recherche permettent de rechercher des covoiturages en fonction des critères suivants : adresse de départ, adresse d'arrivée et date de départ. SearchCarpooling() // SQL request, pas besoin de tout récupérer

Les covoiturages sélectionnés s'affichent dans un tableau après que l'utilisateur ait saisi au moins un critère. La liste s'affine ensuite au fur et à mesure que les autres critères sont saisis.

Les covoiturages n'ayant plus de places disponibles sont affichés et les boutons Réserver ne sont pas accessibles pour cette catégorie. Calculer le nombre de places restantes  
CalculateLeftPlaces(Vehicle vehicle, Reservation reservation)

Je peux effectuer les actions suivantes:

- Un clic sur le bouton Réserver permet d'afficher une fenêtre de confirmation. La fenêtre de confirmation contient les informations du covoiturage (véhicule et la liste des passagers).
  - Un clic sur le bouton Annuler, ferme la fenêtre GetCarpoolingPassengers(Reservation reservation)
  - Un clic sur le bouton Confirmer, réserve une place dans le covoiturage.



### **Annuler ma participation à un covoiturage**

`CancelCarpoolingBooking()`

En tant que collaborateur je peux annuler ma participation à un covoiturage.

Une fenêtre de confirmation s'ouvre avec le détail du covoiturage.

Je peux effectuer les actions suivantes:

- Un clic sur le bouton Annuler, ferme la fenêtre
- Un clic sur le bouton Confirmer, annule ma place dans le covoiturage.



### **Lister ses annonces de covoiturage dont je suis l'organisateur**

En tant que collaborateur, je souhaite pouvoir visualiser les annonces de covoiturage dont je suis l'organisateur et donc le chauffeur.

`GetAllMyOrganizedCarpoolings()`

Je peux visualiser dans un premier tableau la liste de mes covoiturages en cours et dans un second tableau la liste de mon historique.

Le tableau affiche les informations suivantes :

- La date heure de départ
- L'adresse de départ
- L'adresse d'arrivée
- Le nombre de voyageurs ayant réservé.

Je peux effectuer les actions suivantes:

- **Modifier** une annonce en cours `UpdateOrganizedCarpooling(id)`
- Annuler une annonce en cours `DeleteOrganizedCarpooling(id)`
- Créer une nouvelle annonce `CreateCarpooling()`

Rajouter une composition sur le diagramme de classes (suppression en cascade après supprimer un covoit)



## Créer une annonce de covoiturage

En tant que collaborateur, je souhaite pouvoir publier une annonce de covoiturage afin de ne pas voyager seul.

Un covoiturage est constitué de :

- Une adresse de départ (saisie assistée facultative avec une API. API Open Street Map)
- Une adresse d'arrivée (idem ci-dessus)
- Une durée de trajet (calculée via une API : calcul facultatif pour ce projet sinon saisie)
- Une distance (calculée via une API : calcul facultatif pour ce projet sinon saisie)
- Un véhicule
  - Une immatriculation
  - Une marque
  - Un modèle
  - Nombre de places disponibles
- Une date/heure de départ

Je peux effectuer les actions suivantes:

- Cliquer sur le bouton publier pour publier l'annonce
- Cliquer sur annuler pour revenir à la liste des covoiturages.



## Modifier une annonce de covoiturage

En tant que collaborateur j'ai la possibilité de modifier une annonce de covoiturage en cours à condition qu'elle ne fasse pas l'objet d'une réservation. Dans ce cas le seul moyen que j'ai est d'annuler l'annonce (voir fonctionnalité suivante).

Règles métier:

- On ne peut modifier une annonce de covoiturage faisant l'objet d'une

réservation Je peux effectuer les actions suivantes:

- valider la modification
- annuler la modification



## Supprimer une annonce de covoiturage

En tant que collaborateur, quel que soit le profil, je peux supprimer une annonce de covoiturage.

Lorsque le collaborateur clique sur le bouton supprimer, une fenêtre de confirmation montre la liste des personnes ayant réservé (le cas échéant) et propose un bouton confirmer et un bouton annuler.

En cas de confirmation, toutes les personnes ayant réservé reçoivent un mail d'avertissement.

NotifyPassengers()

A partir de cette liste je peux effectuer les actions suivantes:

- Confirmer ma suppression
- Annuler ma suppression

GetCarpoolingPassengers(Reservation reservation)

→ nécessite une vérification avant la modification.



## Lister les réservations de véhicule de service

`GetAllMyVehiclesReservations()`

En tant que collaborateur j'ai la possibilité de visualiser mes réservations en cours et historique de véhicules de service.

Un tableau présente la liste des réservations en cours et un second la liste des réservations historique.

Chaque tableau présente les informations suivantes :

- Date heure de début
- Date heure de fin
- Immatriculation
- Marque
- Modèle

Je peux effectuer les actions suivantes :

- Modifier une réservation en cours `UpdateVehicleReservation()`
- Annuler une demande en cours `DeleteVehicleReservation()`
- Réaliser une nouvelle réservation `CreateVehicleReservation()`



## Réservation d'un véhicule de service

En tant que collaborateur j'ai la possibilité de réserver un véhicule de service.

La réservation est constituée de :

- Une date heure de départ
- Une date heure de retour estimée du véhicule
- Un véhicule de service sélectionné via un slideshow (carrousel) qui affiche la photo du véhicule, son immatriculation, sa marque, son modèle, son nombre de place

Règles métier :

**GetAvailableVehicles()**

- Seuls les véhicules disponibles (sur critères de réservation ou de travaux sur le véhicule) aux dates saisies sont présentés.

Je peux effectuer les actions suivantes:

- Cliquer sur le bouton réserver pour réserver le véhicule
- annuler la création auquel cas je reviens à la visualisation des réservations



## Modifier la réservation d'un véhicule de service

En tant que collaborateur j'ai la possibilité de modifier ma réservation d'un véhicule de société.

Règles métier :

- Possibilité de changer n'importe quelle information **ValidateVehicleReservationDates()**
- Attention si je change les dates, vérifier que le véhicule n'a pas été réservé aux nouvelles dates. **Vérifier les dates avant la validation de la modification d'une reservation véhicules**

Je peux effectuer les actions suivantes:

- Cliquer sur le bouton confirmer pour valider ma modification
- annuler la modification auquel cas je reviens à la visualisation des réservations



## Supprimer la réservation d'un véhicule de service

En tant que collaborateur j'ai la possibilité de supprimer ma réservation d'un véhicule de service.

Règles métier :

- RAS

Je peux effectuer les actions suivantes:

- Cliquer sur le bouton confirmer pour valider mon annulation
- annuler la l'annulation auquel cas je reviens à la visualisation des réservations

javascript

← Fenvoie b90p



## Visualiser les véhicules de service

En tant qu'administrateur, je souhaite pouvoir accéder à la liste des véhicules de service afin de pouvoir les gérer.

Pour chaque véhicule, les informations suivantes sont affichées :

- Immatriculation
- Marque
- Modèle
- Catégorie
  - Micro-urbaines
  - Mini-citadines
  - Citadines polyvalentes
  - Compactes
  - Berlines Taille S
  - Berlines Taille M
  - Berlines Taille L
  - SUV, Tout-terrains et Pick-up
- Photo (URL)
- Motorisation (Hybride, électrique, etc.)
- CO2/km
- Nombre de places (chauffeur inclus)

Je peux effectuer les actions suivantes:

- Il est possible de filtrer la liste par Immatriculation ou par marque. `GetVehicules(filter)`
- Ajouter un véhicule. `CreateVehicle()`
- Modifier un véhicule `UpdateVehicle(id)`
- Supprimer un véhicule `DeleteVehicle(id)`





## Saisir un nouveau véhicule de service

En tant qu'administrateur je peux ajouter un véhicule de service.

Les informations à saisir sont les suivantes :

- Immatriculation
- Marque
- Modèle
- Catégorie
  - Micro-urbaines
  - Mini-citadines
  - Citadines polyvalentes
  - Compactes
  - Berlines Taille S
  - Berlines Taille M
  - Berlines Taille L
  - SUV, Tout-terrains et Pick-up
- Photo
- Nombre de places
- Motorisation (Hybride, électrique, etc..)
- CO2/km
- Statut (en service obligatoirement)

Règles métier:

- La photo n'est pas stockée dans l'application, il s'agit d'une URL qui accessible en ligne.

A partir de cette liste de je peux effectuer les actions suivantes:

- Valider le nouveau véhicule de service **Ne pas dupliquer un véhicule (immatriculation)**
- Annuler l'ajout ce qui fait revenir à la visualisation des véhicules



## Modifier un véhicule de services et gérer son cycle de vie

En tant qu'administrateur je peux **modifier** un véhicule de service.

Les informations modifiables sont les suivantes :

- Immatriculation
- Marque
- Modèle
- Catégorie
- Photo
- Nombre de places
- Motorisation (Hybride, électrique, etc..)
- CO2/km
- Statut (en service, en réparation ou hors service)

**GetAllVehicleReservations()**

J'ai accès également à un **tableau récapitulatif des réservations en cours, futures et passées.**

Chacun de ces tableaux présente les informations suivantes :

- Date heure de début
- Date heure de fin
- Responsable (personne ayant effectuée la réservation)

Règles métier:

- La photo n'est pas stockée dans l'application, **il s'agit d'une URL qui accessible en ligne.**
- Si un véhicule est mis **hors service ou en réparation**, les réservations effectuées pour ce véhicule sont annulées et **un mail est envoyé aux responsables.**

**UpdateVehicleStatus()**

**NotifyPassengers()**

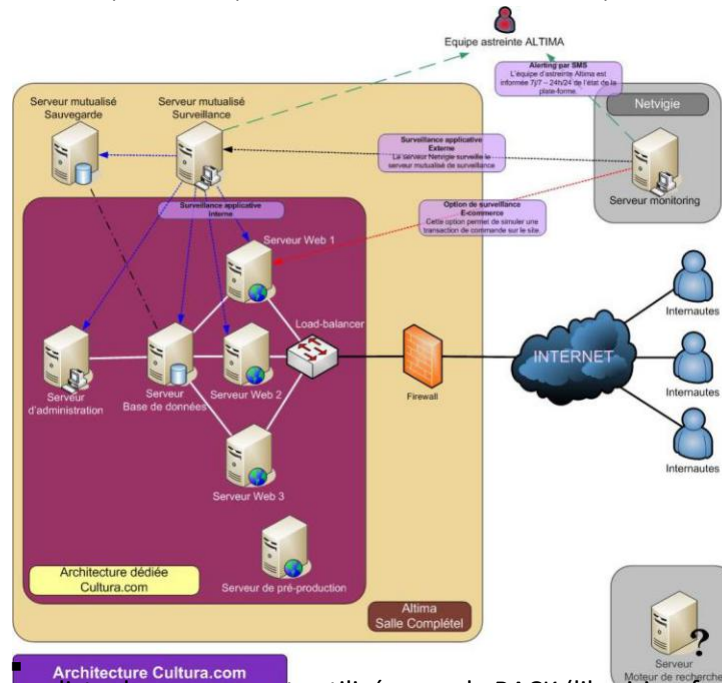
A partir de cette liste de je peux effectuer les actions suivantes:

- Valider la modification du véhicule de service
- Annuler la modification ce qui fait revenir à la visualisation des véhicules

## Travail à réaliser :

- Bien comprendre le besoin – échanges avec le PO
  - Définir avec le client toutes les règles de gestion (règles métier)
  - Préparation du backlog piloté par le PO
- Réaliser les maquettes
- Rédiger les spécifications fonctionnelles générales.
  - Prioriser les divers cas d'utilisation avec le PO
  - Réaliser les diagrammes de cas d'utilisation avec les acteurs associés
  - Pour chaque fonctionnalité :
    - Reformuler le besoin du Client en précisant toutes les règles de gestion identifiées.
    - Afficher la copie d'écran de la maquette
    - Détailler l'ensemble des actions réalisables par l'utilisateur et ce que cela produit
    - Détailler les exceptions à afficher lorsque l'utilisateur réalise une action qui enfreint une règle de gestion.
- Rédiger les spécifications générales détaillées (dossier de conception)
  - Architecture technique

Exemple de description de la future architecture technique :



- Architecture Cultura.com
  - liste des composants utilisés pour le BACK (bibliothèques, frameworks)
  - liste des composants utilisés pour le FRONT
  - liste des logiciels utilisés sur les postes de développement :
    - version de langage
    - base de données utilisée
    - serveur de déploiement
  - Quelle architecture pour le développement : hébergement du code source ?
- Avec Docker
- Modèle logique de données (diagramme de classes)
- Modèle physique de données

- Règles de développement :
  - découpage en couche
  - bonnes pratiques à respecter
- Diagramme de classes avec les classes techniques  
exemple type
- Diagrammes de séquences pour le traitement de nuit
- *Facultatif : Description des échanges clients/serveurs entre back et front*
- *Facultatif : Décrire l'API avec les URL et les verbes HTTP*