

# CAHIER DES CHARGES



Lien TRELLO : <https://trello.com/b/jdyXm3Ws/conduite-de-projet>

Lien github côté Front : <https://github.com/Michelhof1978/GarageParrot>

Lien github côté back : <https://github.com/Michelhof1978/GarageBack>

Lien Figma : <https://www.figma.com/file/rhs91pTc1st89ApCbMHoel/GARAGE-PARROT?type=design&node-id=0-1&mode=design&t=b8DACnrXU32mmG6U-0>

## **Documents joints avec le projet :**

- Maquette Figma
- User stories admin + Utilisateurs
- Cahier des charges
- Gestion des tâches avec Trello
- Diagramme de cas d'utilisation
- Diagramme de séquences
- Diagramme de classes avec Mysql

## **TECHNOLOGIES UTILISEES POUR LE PROJET**

- Front : React, Bootstrap, Html et Css
- Back : Php 8, MySqlworkbench et Xampp

## **DESCRIPTION DU PROJET**

Le projet vise à développer une application web vitrine pour le [Garage V. Parrot](#), un établissement automobile situé à Toulouse, qui propose une gamme de services incluant la réparation automobile, l'entretien et la vente de véhicules d'occasion. L'objectif principal de cette application est de mettre en avant la qualité des services offerts par le garage.

Voici un aperçu des principales fonctionnalités à implémenter :

**1. Connexion :** L'application doit permettre à l'administrateur et aux employés de se connecter via un formulaire d'accès à l'espace professionnel. Les identifiants requis seront le nom d'utilisateur et un mot de passe sécurisé. Seul l'administrateur principal sera autorisé à créer, modifier ou supprimer des comptes pour ses employés. Dans le tableau de bord, toutes les sections seront accessibles à tous les utilisateurs, à l'exception de la section "gestion des comptes employés", qui sera réservée à l'administrateur en chef pour gérer les comptes de ses employés.

**2. Présentation des services :** La page d'accueil de l'application devra clairement présenter les différents services de réparation automobile proposés par le garage, tant dans la barre de navigation que sous forme de diaporama sur la page d'accueil. L'administrateur devra avoir la capacité d'ajouter, de modifier ou de supprimer ces informations depuis son espace d'administration. De plus, seul le compte administrateur pourra consulter l'historique des modifications apportées par les employés, afin de retracer toute manipulation effectuée par un employé en cas de problème.

**3. Définir les horaires d'ouverture :** Les horaires d'ouverture du garage seront clairement affichés sur le site, de préférence dans le pied de page. L'administrateur aura la possibilité de spécifier ces horaires depuis son espace dédié. Les horaires seront affichés avec une coupure entre [12h et 14h](#) pour indiquer les heures de fermeture pendant cette période.

**4. Présentation des voitures d'occasion :** L'application devra afficher les véhicules disponibles à la vente, accompagnés d'une galerie de photos, de descriptions détaillées et d'informations techniques telles qu'un tableau de caractéristiques, la liste des équipements et options installés. Chaque véhicule devra présenter son prix, une image principale, l'année de mise en circulation et le kilométrage. Les employés auront la possibilité d'ajouter de nouvelles voitures depuis leur espace dédié. Chaque nouvelle voiture ajoutée sera automatiquement mise en avant sur la page d'accueil sous forme de carte, permettant ainsi de présenter les nouveautés aux utilisateurs. Les voitures seront triées automatiquement par ordre d'arrivée.

**5. Filtrage de la liste des véhicules d'occasion :** Pour simplifier la recherche des visiteurs, un système de filtres sera mis en place. Les visiteurs pourront affiner leurs résultats en fonction d'une fourchette de prix ([slider](#)), d'un nombre de kilomètres parcourus ou de l'année de mise en circulation. Pour offrir la meilleure expérience utilisateur possible, les filtres seront appliqués en temps réel, [sans nécessiter le rechargement de la page](#). Les voitures seront classées en cinq catégories : [Berline](#), [SUV](#),

Familiale, Utilitaire et Citadine, avec des options de mise en avant sous forme de boutons (cases à cocher) pour une expérience utilisateur optimale.

**6. Contact avec l'atelier :** Les visiteurs devront avoir la possibilité de contacter facilement le garage, que ce soit par téléphone ou en remplissant un formulaire de contact qui sera ensuite redirigé vers le service approprié. L'utilisateur devra fournir son nom, prénom, adresse e-mail, numéro de téléphone et un message. De plus, une liste déroulante permettra de choisir vers quel service l'utilisateur souhaite diriger sa demande. En cas de contact concernant un véhicule en vente, l'ID ou la référence du véhicule et la fiche du véhicule seront automatiquement inclus dans le formulaire, accompagnés du message de l'utilisateur. Cela permettra à l'administrateur de retrouver la voiture plus facilement et de répondre de manière plus efficace à la demande de l'utilisateur, ce qui permettra de gagner du temps. L'application intégrera également des mécanismes de conformité au RGPD pour la protection des informations personnelles et un captcha de Google pour renforcer la sécurité. pour prouver que c'est un humain et éviter que le message tombe dans les spams. Un message s'ouvrira dans une autre page pour confirmation d'envois.

**7. Collecte des témoignages clients :** Une section dédiée sera créée pour permettre aux visiteurs de laisser des témoignages. Ces témoignages devront être soumis à une modération préalable par un employé du garage avant d'être affichés sur la page d'accueil. Les employés auront également la possibilité de laisser des avis client via l'espace Pro. Un système de notation à cinq étoiles sera mis en place, permettant aux utilisateurs de sélectionner leur note, d'ajouter leur nom et de laisser un commentaire. Les informations personnelles ne seront pas affichées publiquement pour préserver la vie privée des utilisateurs. Diffusion des avis sous forme de carrousel.

## 8. Affichage :

-Cookies (Acceptation RGPD, affichage page d'accueil lors de la première connexion)

Principales utilisations des cookies :

1. Maintenir la session de l'utilisateur
2. Améliorer l'expérience utilisateur
3. Suivi de l'activité et analyse
4. Publicité ciblée
5. Mesure du trafic

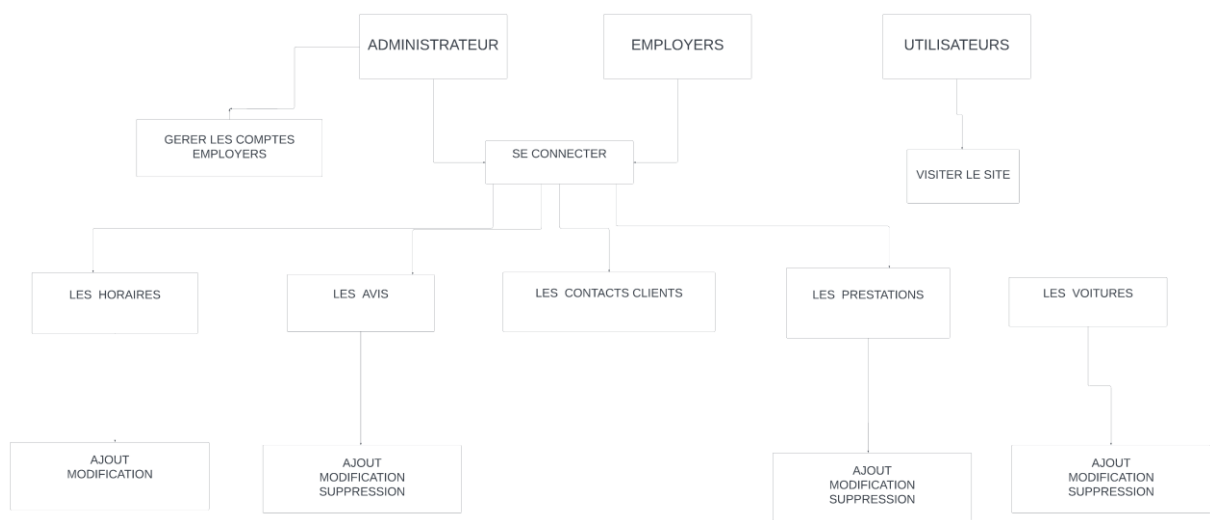
-Mentions légales

-Condition générales de ventes

-Politique de confidentialité

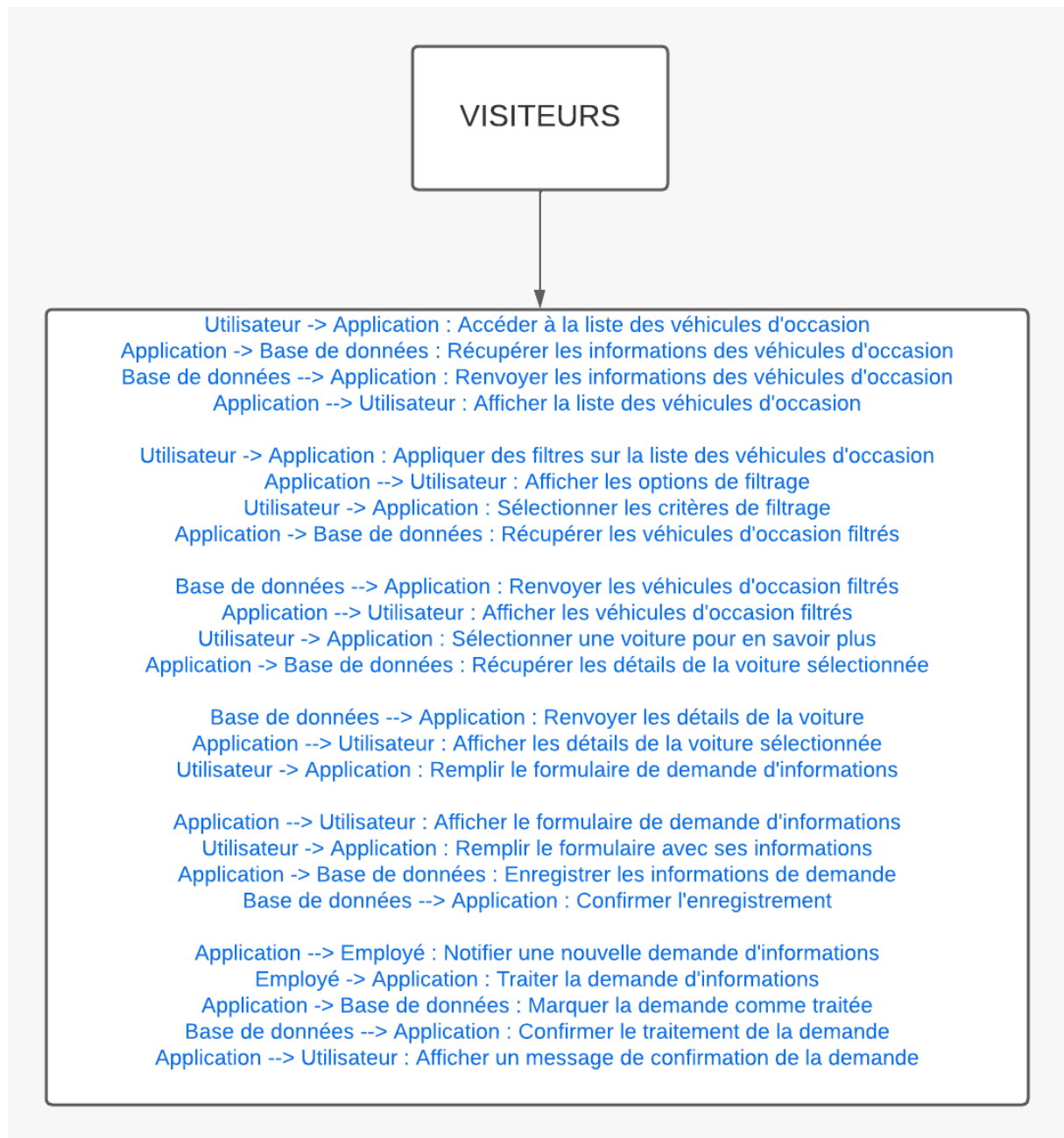
# DIAGRAMMES

## DIAGRAMME DE CAS D'UTILISATION



## DIAGRAMMES DE SEQUENCES

Ces diagrammes de modélisation utilisés représentent les interactions entre différents objets ou acteurs dans l'application. Ils permettent de visualiser la séquence des messages échangés entre ces objets ou acteurs dans le temps. Ils sont particulièrement utiles pour décrire le flux de contrôle entre les différentes parties d'un système et pour comprendre le comportement dynamique d'un processus comme ci-dessous.



Ce diagramme de séquence montre comment un nouveau visiteur peut découvrir la liste des véhicules d'occasion, appliquer des filtres pour affiner les résultats, sélectionner une voiture spécifique et remplir le formulaire de demande d'informations pour en savoir plus sur cette voiture. L'application enregistre ensuite la demande et la notifie à un employé du garage qui traitera la demande et marquera la demande comme traitée dans la base de données.

## Administrateur

Administrateur -> Application : Se connecter en tant qu'administrateur  
Application -> Base de données : Vérifier les identifiants de l'administrateur  
Base de données --> Application : Renvoyer les informations d'identification de l'administrateur

Administrateur -> Application : Accéder à l'espace d'administration  
Application --> Administrateur : Afficher l'espace d'administration

Administrateur -> Application : Gérer les services de réparation automobile  
Application -> Base de données : Récupérer les informations des services de réparation  
Base de données --> Application : Renvoyer les informations des services de réparation  
Application --> Administrateur : Afficher les services de réparation

Administrateur -> Application : Ajouter un nouveau service de réparation  
Application --> Administrateur : Afficher le formulaire d'ajout de service  
Administrateur -> Application : Remplir les informations du nouveau service  
Application -> Base de données : Enregistrer le nouveau service  
Base de données --> Application : Confirmer l'enregistrement du nouveau service  
Application --> Administrateur : Afficher un message de confirmation

Administrateur -> Application : Modifier un service de réparation existant  
Application --> Administrateur : Afficher le formulaire de modification de service  
Administrateur -> Application : Modifier les informations du service  
Application -> Base de données : Mettre à jour les informations du service  
Base de données --> Application : Confirmer la mise à jour du service  
Application --> Administrateur : Afficher un message de confirmation

Administrateur -> Application : Supprimer un service de réparation existant  
Application --> Administrateur : Afficher la confirmation de suppression du service  
Administrateur -> Application : Confirmer la suppression du service  
Application -> Base de données : Supprimer le service de réparation  
Base de données --> Application : Confirmer la suppression du service  
Application --> Administrateur : Afficher un message de confirmation

Administrateur -> Application : Gérer les horaires d'ouverture  
Application -> Base de données : Récupérer les horaires d'ouverture  
Base de données --> Application : Renvoyer les horaires d'ouverture  
Application --> Administrateur : Afficher les horaires d'ouverture

Administrateur -> Application : Modifier les horaires d'ouverture  
Application --> Administrateur : Afficher le formulaire de modification des horaires  
Administrateur -> Application : Modifier les horaires d'ouverture  
Application -> Base de données : Mettre à jour les horaires d'ouverture  
Base de données --> Application : Confirmer la mise à jour des horaires  
Application --> Administrateur : Afficher un message de confirmation

Administrateur -> Application : Gérer les témoignages clients  
Application -> Base de données : Récupérer les témoignages clients  
Base de données --> Application : Renvoyer les témoignages clients  
Application --> Administrateur : Afficher les témoignages clients

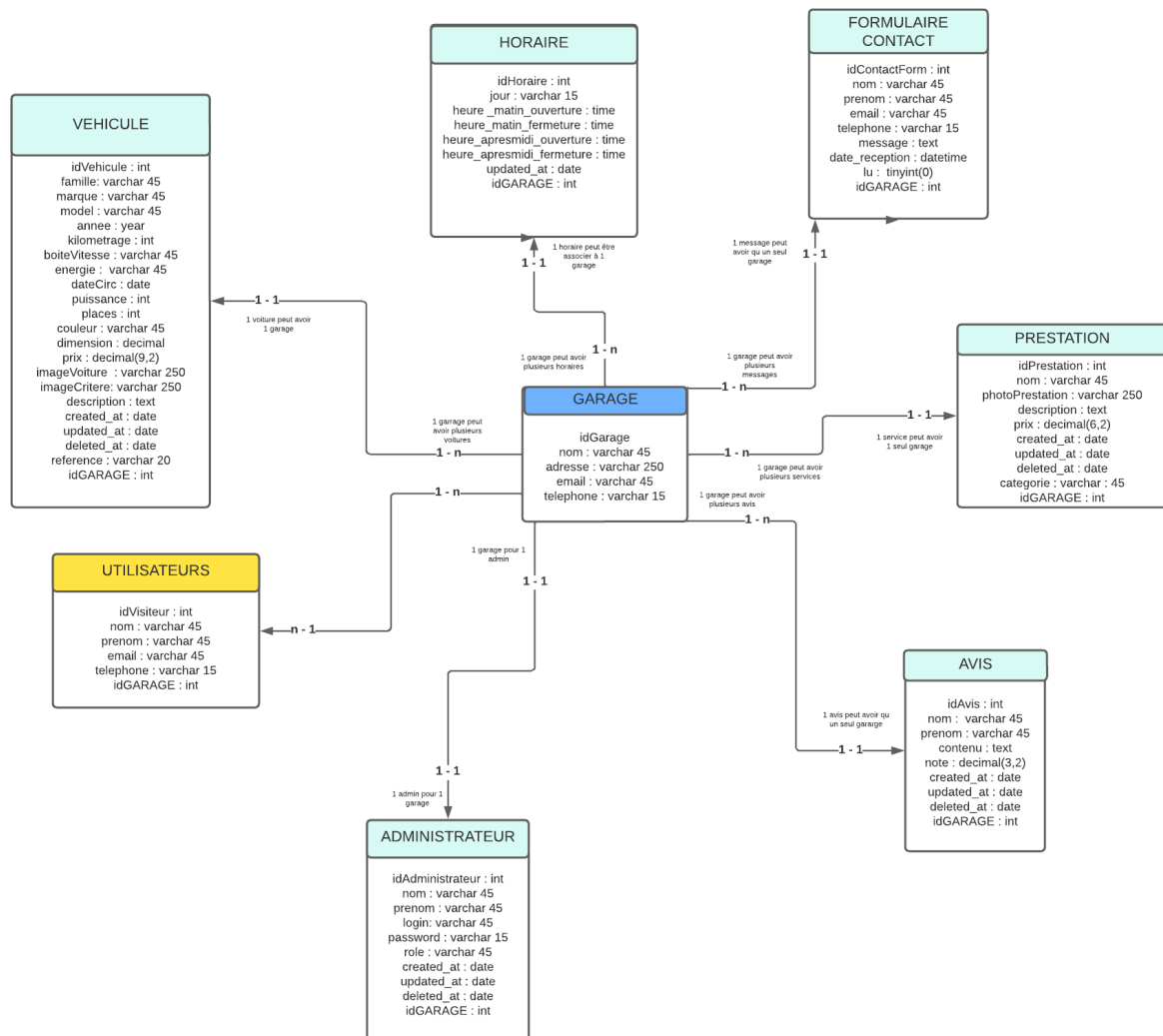
Administrateur -> Application : Modérer un témoignage client  
Application --> Administrateur : Afficher le formulaire de modération de témoignage  
Administrateur -> Application : Modifier l'état du témoignage (approuvé/rejeté)  
Application -> Base de données : Mettre à jour l'état du témoignage  
Base de données --> Application : Confirmer la mise à jour de l'état du témoignage  
Application --> Administrateur : Afficher un message de confirmation

Ce diagramme de séquence montre comment un administrateur peut se connecter à l'application, accéder à l'espace d'administration, gérer les services de réparation automobile (ajouter, modifier, supprimer), gérer les horaires d'ouverture, et modérer les témoignages clients. L'application interagit avec la base de données pour récupérer et mettre à jour les informations pertinentes, et elle affiche des messages de confirmation à l'administrateur pour indiquer le succès des opérations effectuées.

## DIAGRAMME DE CLASSES

Ce diagramme de classes illustre la structure de base du modèle de données pour le

### Garage V. Parrot



-Dans ce diagramme de classes, le Garage est la classe principale et possédera des méthodes pour gérer les services, les horaires, véhicules d'occasion, les contacts avec le service client ainsi que les avis clients.

-La classe Service représente un service de réparation automobile. Elle a des méthodes pour accéder et modifier ces attributs.

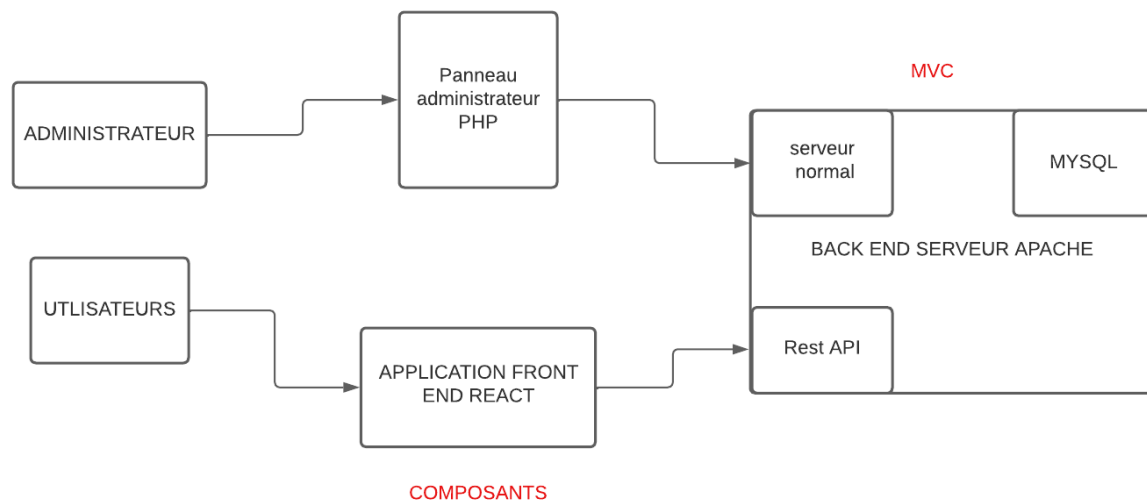
-La classe Véhicule représente un véhicule d'occasion. Elle a des méthodes pour accéder et modifier ces attributs.

-La classe Message représente les informations de contact, telles que les demandes de contact de clients.

-La classe Horaire représente les horaire d'ouverture du garage . Elle peut avoir des méthodes pour accéder et modifier ces attributs.

-La classe Avis représente les avis ou les témoignages des clients concernant le garage ou ses services. Elle peut avoir des méthodes pour accéder et modifier ces attributs, ainsi que pour gérer les opérations liées aux avis.

## CONCEPTION ET ARCHITECTURE



Le projet contiendra 2 parties distinctes :

-La **partie back end** qui sera réalisé en php et placé sur un serveur Apache avec une base de données mysql/apache.

-Le **panneau administrateur** alimentateur et générera cette base de données en PHP. Toute la partie Serveur utilisera l'architecture modèle du contrôleur et sera programmée en POO.

-L'utilisateur utilisera les données de l'**API REST PHP** qui devra être programmé au niveau serveur et qui seront en format **JSON**.

Le serveur devra renvoyer uniquement les données en **Json** et ça sera **React** qui se chargera de les afficher.



## CODE REFERENCE VEHICULE

Référence id unique par véhicule : exemple = **P-C-208-année**

Code référence véhicule par **marque**:

Code référence par **famille** :

**Peugeot = P**

**Renault = R**

**Citroen = C**

**Wolkswagen = w**

**Fiat = F**

**Utilitaire = U**

**Berline = B**

**familiale = F**

**Citadine = C**

**Suv = S**

Code référence véhicule par marque/model

**Peugeot = 208/405/2008/3008/5008**

**Renault = clio/megane/zoe/twingo/kadjar**

**Citroen = C3/C4/C5/berlingo/spacetourer**

**Wolkswagen = golf/polo/passat/tiguan/touareg**

**Fiat = 500/panda/ducato/punto/tipo**

## PALETTE DE COULEURS

	#04BBFF
	#04BBFF
	#04BBFF
	#003C57
	#003C57

## POLICES

- **LEAD** de Bootstrap

- LEAD n'est pas reconnu par Figma, je mettrai en remplacement sur la maquette la police

'Time New Roman'

-Unkempt