

# PROJET CAR'APP

PWEB – 2020

## Sommaire

I – Introduction .....	2
II – Aspects importants de l'application.....	2
III – Liste des services opérationnels .....	3
IV – Architecture et structure du code .....	15
V – Schéma relationnel de la base et explications.....	16
VI – Outils.....	20
VII – Bilan du projet .....	21

DOUMECHE Jules

MARTIN Gwénolé

HACHISMAN Mehdi

Encadrement : ILIE Jean-Michel

## I – Introduction

Pour ce projet de Programmation Web, nous avons réalisé une application web de gestion de locations de voitures sous Symfony. Cette application permet à des loueurs de proposer des voitures à des clients, qui pourront louer automatiquement les voitures via le site.

Toutes les informations nécessaires à l’installation du projet sont disponibles dans le fichier Readme.

Ce projet est disponible dans son intégralité sur Github : <https://github.com/julio4/CarApp>

## II – Aspects importants de l’application

Il y a 3 types d’utilisateurs sur l’application, le rôle administrateur permet de définir les rôles des autres utilisateurs (loueurs ou clients). Les changements se font via le panel administrateur.

Le Loueur a accès à un panel loueur riche, lui permettant d’ajouter, modifier ou supprimer des voitures et d’ajouter ou modifier des types. Il peut aussi voir ses dernières locations, la liste de ses clients, la liste de tous ses véhicules et d’autres informations.

Le loueur peut aussi afficher la facturation de chaque client, ce qui lui permet de savoir combien chaque client lui doit et ce qui a déjà été payé.

Le client peut lui louer des voitures sur le site, et voir dans son panel la liste de ses locations. Il peut depuis son panel payer les locations et voir les détails. Il peut aussi obtenir le coût total de ses locations par mois.

Chaque service va être expliqué en détail ci-dessous. Les services qui fonctionnaient lors du premier rendu étaient les services liés à l’administrateur, la création/connexion à un compte, l’ajout d’un véhicule et la location d’un véhicule.

Pour se connecter en mode admin, utiliser *admin*, mot de passe *password*

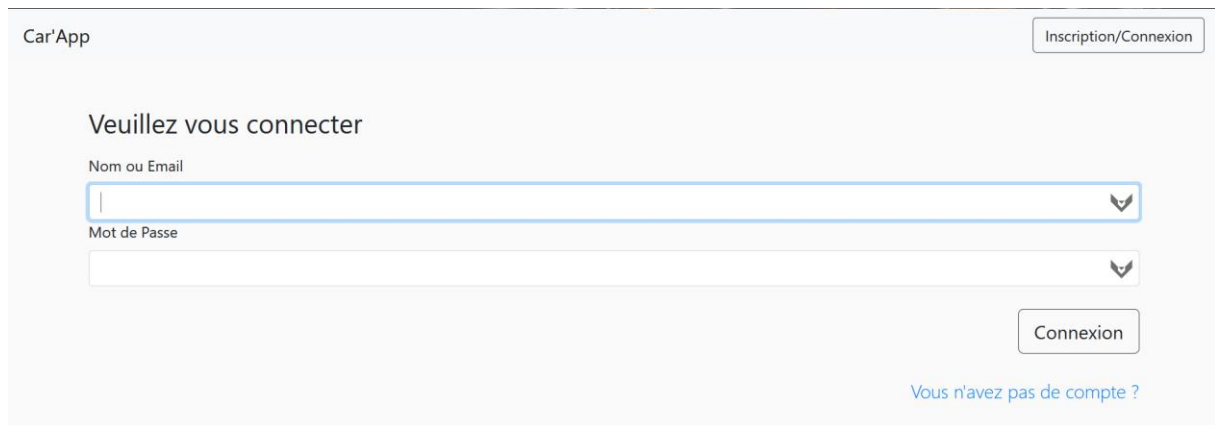
Pour se connecter en mode loueur, utiliser *loueur*, mot de passe *password*

Pour se connecter avec les comptes clients prédéfinis, le mot de passe est toujours *passwordpassword* et il faut utiliser l’adresse mail ou le nom complet

### III – Liste des services opérationnels

#### Service sans compte

- Inscription et connexion des clients et des loueurs



Car'App Inscription/Connexion

Veuillez vous connecter

Nom ou Email

Mot de Passe

Connexion

[Vous n'avez pas de compte ?](#)



Car'App Inscription/Connexion

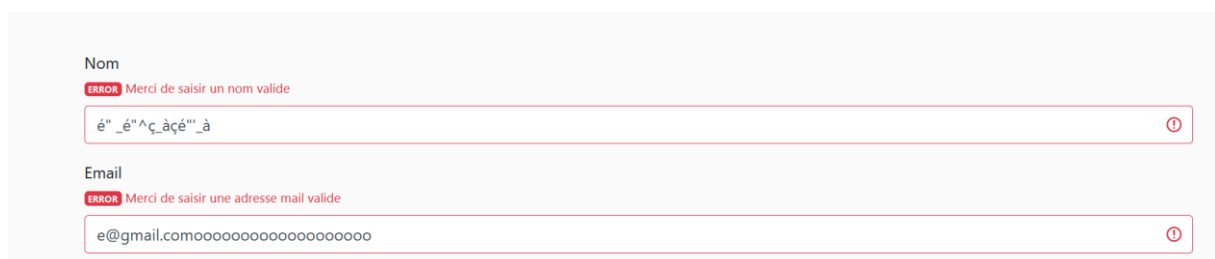
Nom

Email

Mot de Passe

Confirmer le mot de passe

S'inscrire



Nom

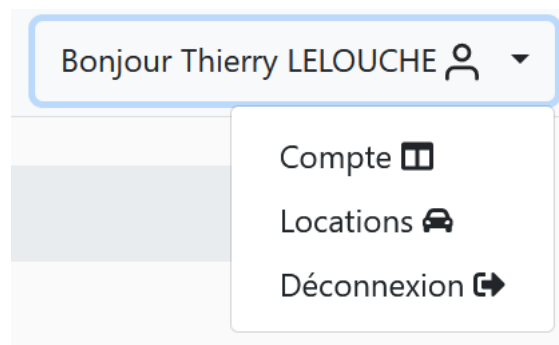
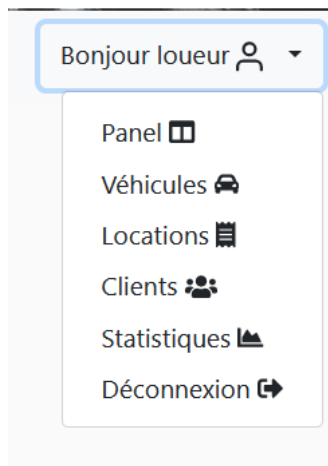
**ERROR** Merci de saisir un nom valide

Email

**ERROR** Merci de saisir une adresse mail valide

Validation errors are shown in red text above the input fields. The first error is for the name field, and the second is for the email field.

Le formulaire d’inscription permet de créer un compte. Ce compte est un compte client par défaut, c’est l’administrateur qui doit passer le compte en rôle loueur. Toutes les saisies sont vérifiées par des Assert sur chaque propriété, et nous utilisons le composant Security pour que les connexions et le stockage des mots de passe soient sécurisées.



## Services avec un compte client

- Location d'un véhicule

La location d'un véhicule s'effectue en deux étapes, il faut au préalable avoir créé un compte client pour ne pas être bloqué plus tard.

The screenshot shows the Car'App homepage. At the top left is the 'Car'App' logo. At the top right is a button labeled 'Inscription/Connexion'. In the center, the text 'Car'App' is displayed in a large font, followed by 'Choisissez parmi nos 12 modèles'. Below this, a prompt says 'Choisissez la date de début de location :'. There is a date selection input field. Below the input field are two buttons: 'Location à des dates précises' (yellow) and 'Valider la date' (blue with a calendar icon). A light blue notification box at the bottom states: 'Vous consultez les véhicules disponibles pour une location mensuelle à durée indéterminée !' with a close icon (X).

Sur la page d'accueil, le client sélectionne une plage de date, ou sélectionne une date de début de location pour une location mensuelle, et clique sur le bouton de validation.

The screenshot shows a grid of eight car models, each with a photo, name, and starting price per month. The cars are: Audi A8 (À partir de 1025 €/mois), BMW Gt6 (À partir de 1161 €/mois), Citroen DS3 (À partir de 594 €/mois), Ferrari 458 Italia (À partir de 5130 €/mois), Ferrari 812 (À partir de 13203 €/mois), Lamborghini Aventador (À partir de 25650 €/mois), Mercedes Benz GLC (À partir de 1620 €/mois >), and Peugeot 208 (À partir de 1134 €/mois). At the bottom of the grid are navigation buttons: '1', '2', and '>'.

Les types de véhicules correspondant à cette sélection s'affichent ensuite, avec une photo et le nom du type. Pour continuer, l'utilisateur doit cliquer sur le type de véhicule qui l'intéresse.

The screenshot shows the date selection screen. At the top, it says 'Choisissez vos dates de location :'. Below this is a date selection input field. At the bottom, there are two buttons: 'Location mensuelle' (teal) and 'Valider les dates' (blue with a calendar icon). A yellow notification box at the bottom states: 'Vous consultez les véhicules disponibles pour une location fixe !' with a close icon (X).

## Rapport de projet PWEB – Car'App



Fenêtre permettant de sélectionner les dates

Une page présente ensuite les différents véhicules disponibles pour le type de véhicule et les dates sélectionnées.



Lamborghini Aventador  
1 available

### Location mensuelle

Un paiement chaque mois, réservé pour vous pour une durée indéterminée, annulable à partir du premier mois!

#### Lamborghini Aventador



**boîte**  
Automatique à simple embrayage

**moteur**  
GPL

**carburant**  
SP98

32535 €/mois

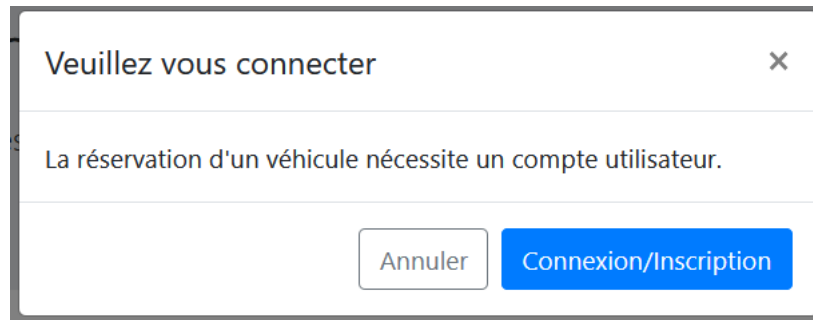
~~3205~~ 1084.5 €/jours -10%

Réserver Maintenant

Le client peut cliquer sur le bouton « Réservez maintenant » pour réserver le véhicule.

Rapport de projet  
PWEB – Car’App

Au moment du clic, si l'utilisateur n'est pas connecté avec un compte client, il lui est demandé de s'en créer un.



A modal dialog box with a white background and a thin gray border. The title bar at the top says "Veuillez vous connecter" in a dark gray font, with a close button (X) on the right. Below the title bar, the text "La réservation d'un véhicule nécessite un compte utilisateur." is displayed in a smaller, lighter gray font. At the bottom, there are two buttons: "Annuler" (white with a gray border) and "Connexion/Inscription" (solid blue).

Si l'utilisateur est bien connecté un message de confirmation s'affiche.



A modal dialog box with a white background and a thin gray border. The title bar at the top says "Confirmation de location" in a dark gray font, with a close button (X) on the right. Below the title bar, the text "Ce véhicule vous sera facturé **14850€/mois**" is displayed in a dark gray font. At the bottom, there are two buttons: "Annuler" (white with a gray border) and "Confirmer" (solid blue).

Une fois validé, un message de confirmation s'affiche en haut de la page ou sur la page d'accueil s'il s'agissait du dernier véhicule de ce type. Ce lien est cliquable et permet à l'utilisateur d'accéder à la page de détails de la commande.

Quand un véhicule est loué pendant une période de temps, il ne peut plus être loué pendant cette période mais peut être loué sur une autre période s'il n'y a pas de chevauchement (sauf les locations mensuelles).

Nouvelle réservation #1 validée à payer.

## Rapport de projet PWEB – Car'App

- Affichage de statistiques sur les locations en cours

The screenshot shows the Car'App dashboard for user Thierry LELOUCHE. The dashboard has a sidebar with 'Accueil', 'Mes Locations', and 'Déconnexion'. The main panel displays two summary cards: 'Véhicules loués' (5) and 'Facture ce mois' (55,464.00 €). Below these is a 'Dernières rentals' section with a table of recent rentals.

#	Modèle	Loueur	Dates	Prix
1	Lamborghini Aventador	loueur	11/13/2020 - 11/13/2020	<b>Paiement mensuel</b> 32535 €/Mois <a href="#">Détails</a>
2	Audi A8	loueur	11/14/2020 - 11/27/2020	840 € <a href="#">Détails</a>
3	Porsche Cayenne	loueur	11/14/2020 - 11/27/2020	4620 € <a href="#">Détails</a>
4	Ferrari 812	loueur	11/13/2020 - 11/13/2020	<b>Paiement mensuel</b> 14850 €/Mois <a href="#">Détails</a>
5	BMW Gt6	loueur	11/13/2020 - 11/13/2020	<b>Paiement mensuel</b> 2619 €/Mois <a href="#">Détails</a>

Le client dispose d'un panel, dont l'accueil lui permet de voir le nombre de véhicules loués, le coût total pour le mois, ainsi que les dernières réservations. En cliquant sur « Tout voir » ou sur « Mes locations », le client accède à la liste complète de ses locations de véhicules.

- Affichage des locations en cours

The screenshot shows the 'Mes Locations' page. At the top, there is a green notification bar: 'Paiement mensuel pour la location #1 effectué avec succès!'. Below this is a table of current rentals with an additional 'Statut' column.

#	Modèle	Loueur	Dates	Prix	Statut
1	Lamborghini Aventador	loueur	13/11/2020	32535 €/Mois <b>Paiement mensuel</b>	<b>Payée ce mois</b> Prochain paiement le 13/12/2020 <b>Stopper la location</b> <a href="#">Détails +</a>
2	Audi A8	loueur	14/11/2020 - 27/11/2020 14 jours	840 €	<b>Payer</b> <a href="#">Détails +</a>
3	Porsche Cayenne	loueur	14/11/2020 - 27/11/2020 14 jours	4620 €	<b>Payée</b> <a href="#">Détails +</a>
4	Ferrari 812	loueur	13/11/2020	14850 €/Mois <b>Paiement mensuel</b>	<b>Payer ce mois</b> <a href="#">Détails +</a>
5	BMW Gt6	loueur	13/11/2020	2619 €/Mois <b>Paiement mensuel</b>	<b>Payer ce mois</b> <a href="#">Détails +</a>

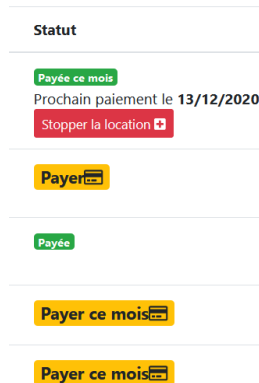
Cette page permet au client de voir tous les véhicules loués. Un clic sur détails redirige vers la page de détail d'une location. Un clic sur « Payer » ou « Payer ce mois » permet au client de payer sa facture.



## Rapport de projet PWEB – Car’App

### - Paiement d’une location

Le clic sur ce bouton permet au client de payer sa location, nous avons volontairement fait un processus automatique pour cette application, mais il est très facile d’implémenter une vraie fonction de paiement avec cette base.



Une fois payée, la location passe en statut « payée ».

Si la durée de location restante est inférieure à 30 jours, le client doit tout payer intégralement.

Si la durée restante est supérieure à 30 jours, le client peut choisir entre payer pour un mois, puis payer plus tard le reste, ou payer l’intégralité.

Si la location est renouvelée mensuellement, le client doit payer une fois par mois.

### - Affichage du détail d’une location

Ce service est aussi disponible depuis le panel loueur, en cliquant sur Détails dans Locations.



Cette page présente l’état d’avancement de la location, elle est créée lors de la confirmation de réservation par un client et mise à jour lors du paiement.

## Services avec un compte loueur

- Création d’un type de véhicule

### Ajouter un nouveau modèle

The screenshot shows a web form titled 'Ajouter un nouveau modèle'. It contains two input fields: 'Nom' with the text 'Fiat Panda' and 'Photo Type du véhicule' which is empty. To the right of the second field is a 'Browse' button. At the bottom center is a blue 'Ajouter' button.

Ce formulaire permet de créer un type. On y accède sur le formulaire de création d’un véhicule, en tapant un nouveau type dans le champ type.

The screenshot shows a dropdown menu with 'Fiat Panda' selected. Below the dropdown is a link that reads 'Ajouter le modèle "Fiat Panda"'. The dropdown is highlighted with a blue border.

Comme tous les formulaires, les champs sont contrôlés.

- Création d’un véhicule

### Ajouter un nouveau véhicule

The screenshot shows a web form titled 'Ajouter un nouveau véhicule'. It features several fields: 'Modèle' (a dropdown menu with 'Audi A8' selected), 'Prix' (a numeric input field with '1,00' and a '€/jour' label), 'Boîte' (a dropdown menu with 'Manuelle' selected), 'Moteur' (a dropdown menu with 'Diesel' selected), 'Carburant' (a dropdown menu with 'Electrique' selected), and 'Photo du véhicule' (an empty input field with a 'Browse' button). At the bottom center is a blue 'Ajouter' button.

## Rapport de projet PWEB – Car’App

Ce formulaire permet d’ajouter un véhicule. On peut y sélectionner diverses caractéristiques et décider de mettre une image ou pas.

Le loueur peut aussi rendre le véhicule disponible immédiatement ou le laisser « non disponible » pour le moment.


Ajouter

☒ Rendre disponible directement ?

### - Modification d’un type de véhicule


Cette page permet de modifier un type de véhicule (le nom ou la photo). C’est le même formulaire que pour la création d’un type et on y accède ici (dans le formulaire de modification d’un véhicule) :

### Modification du véhicule



Modèle

Audi A8

Modifier 

1 semaine : 378.00 €

1 mois : 1360.80 €

1 an: 16329.60 €

## Rapport de projet PWEB – Car'App

- Modification et suppression d'un véhicule

The screenshot shows a web form for modifying a vehicle. At the top, there's a back arrow and a 'Modèle' dropdown menu set to 'Audi A8'. Below this is a 'Prix' field with a minus button, a value of '60,00', and a plus button with the unit '€/jour'. A button labeled 'Estimation des revenus' is positioned below the price field. The form includes three dropdown menus: 'Boite' (set to 'Manuelle'), 'Moteur' (set to 'Diesel'), and 'Carburant' (set to 'Electrique'). A 'Photo du véhicule' section features a 'Supprimer' checkbox, a 'Browse' button, and a photo of a silver Audi A8. Below the photo is a 'Télécharger' link. A blue button 'Confirmer les modifications' is centered below the photo. At the bottom, there's a toggle switch for 'Disponible à la location' (which is turned on) and a red button 'Retirer le véhicule'.

Cette page permet de modifier un véhicule, il est possible d'y accéder via le bouton modifier sur la page des véhicules du panel loueur. Elle ressemble beaucoup au formulaire d'ajout d'un véhicule. Il est aussi possible de supprimer le véhicule sur cette page.

A noter qu'il n'est pas possible d'accéder à cette page si le véhicule est en cours de location par un client (le bouton d'accès est grisé).

- Affichage des statistiques du parc locatif

The screenshot shows the 'Car'App dashboard for a user named 'Bonjour loueur'. The dashboard has a sidebar with navigation links: 'Accueil', 'Véhicules', 'Locations', 'Clients', and 'Déconnexion'. The main area is titled 'Panel' and contains three summary cards: 'Revenus' (42528 €), 'Réservations' (8), and 'Clients' (2). Below these cards is a section titled 'Dernières réservations' containing a table of recent reservations.

#	Modèle	Client	Montant	
2	Audi A8	thierry.lelouche@gmail.com	840 €	Détails
3	Porsche Cayenne	thierry.lelouche@gmail.com	4620 €	Détails
8	BMW Gt6	charles.henri@gmail.com	1809 €	Détails
4	Ferrari 812	thierry.lelouche@gmail.com	14850 €	Détails

La page d'accueil du panel loueur présente quelques statistiques sur le parc locatif du loueur.

La liste des dernières locations est aussi disponible.

## Rapport de projet PWEB – Car’App

### - Affichage de la liste des véhicules

#7	Audi A8	<ul style="list-style-type: none"><li>boite : Automatique à double embrayage</li><li>moteur : Essence</li><li>carburant : SP98</li></ul>	120 €	Disponible	Modifier
#8	Audi A8	<ul style="list-style-type: none"><li>boite : Automatique à double embrayage</li><li>moteur : Essence</li><li>carburant : SP95-E10</li></ul>	38 €	Disponible	Modifier
#9	BMW Gt6	<ul style="list-style-type: none"><li>boite : Automatique à convertisseur de couple</li><li>moteur : Essence</li><li>carburant : GPLc</li></ul>	43 €	Louée	En location
#10	BMW Gt6	<ul style="list-style-type: none"><li>boite : Automatique à double embrayage</li><li>moteur : Essence</li><li>carburant : SP95-E10</li></ul>	67 €	Non disponible	Modifier

Cette page présente la liste complète des véhicules du loueur. Quand le loueur clique sur un véhicule il accède à la page de modification de ce véhicule, ce qui peut lui permet de changer son statut à indisponible par exemple.

### - Affichage des locations en cours

Panel / Locations					
Locations en cours					
#	Modèle	Dates	Montant	Paieement	Statut
2	Audi A8	11/14/2020 - 11/27/2020	840 €	En attente du paiement	Réservé
3	Porsche Cayenne	11/14/2020 - 11/27/2020	4620 €	Payée	Réservé
8	BMW Gt6	11/13/2020 Mensuel	1809 €/mois	En attente du paiement	En cours
4	Ferrari 812	11/13/2020 Mensuel	14850 €/mois	En attente du paiement	En cours
5	BMW Gt6	11/13/2020 Mensuel	2619 €/mois	En attente du paiement	En cours

Cette page affiche toutes les locations en cours. Le clic sur le bouton « Détails » permet d’accéder à la page de détails de la location (la même que celle du client).

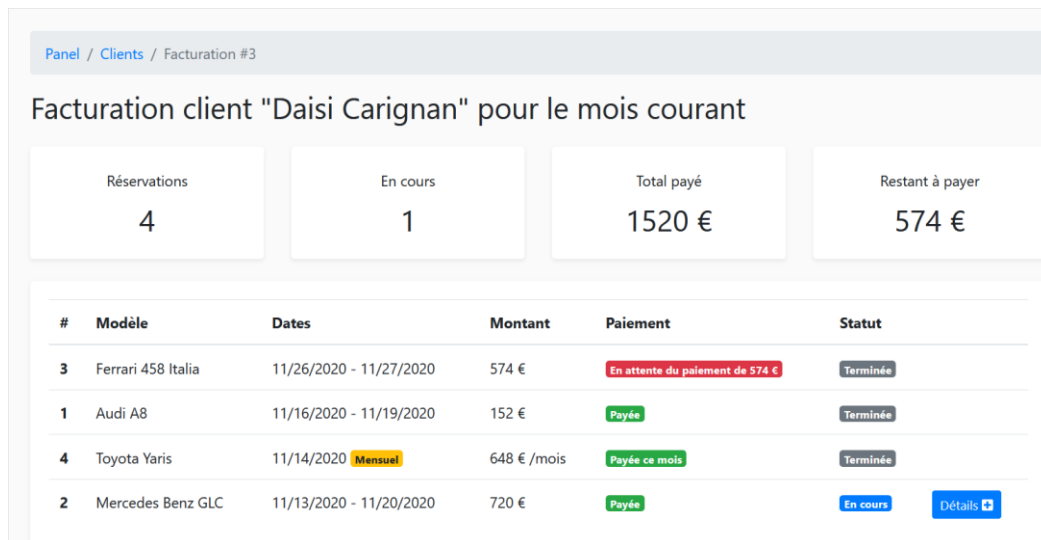
### - Affichage des clients

Clients		
Nom	email	Facturation
charles	charles.henri@gmail.com	Accéder à la facture
Thierry LELOUCHE	thierry.lelouche@gmail.com	Accéder à la facture

Cette page permet au loueur d’accéder à la liste de tous ses clients et d’avoir accès aux factures de chaque client. Le clic sur « Accéder à la facture » permet d’accéder à la page détaillée de la facturation du client.

## Rapport de projet PWEB – Car’App

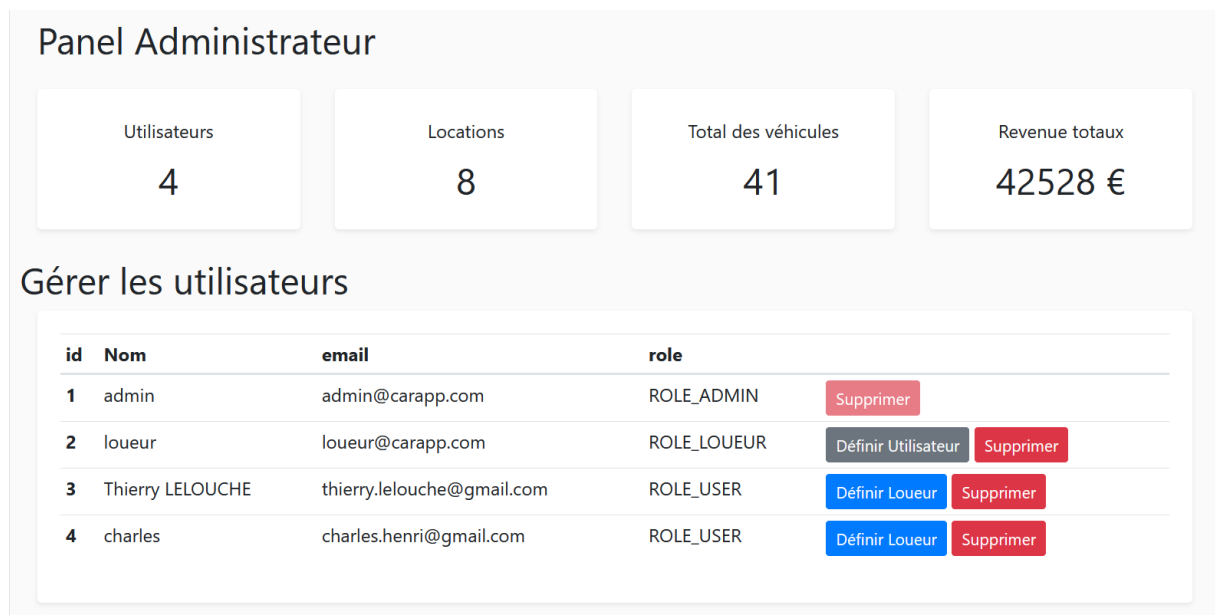
- Affichage détaillé de chaque client, avec ses véhicules pour la facturation



Cette page permet au loueur de savoir exactement ce que lui doit un client. On y retrouve les informations sur chaque véhicule loué par le client, le prix total, le nombre de véhicules et d’autres informations pratiques.

### Services avec un compte administrateur

- Affichage des statistiques globales de l’application et gestion des utilisateurs



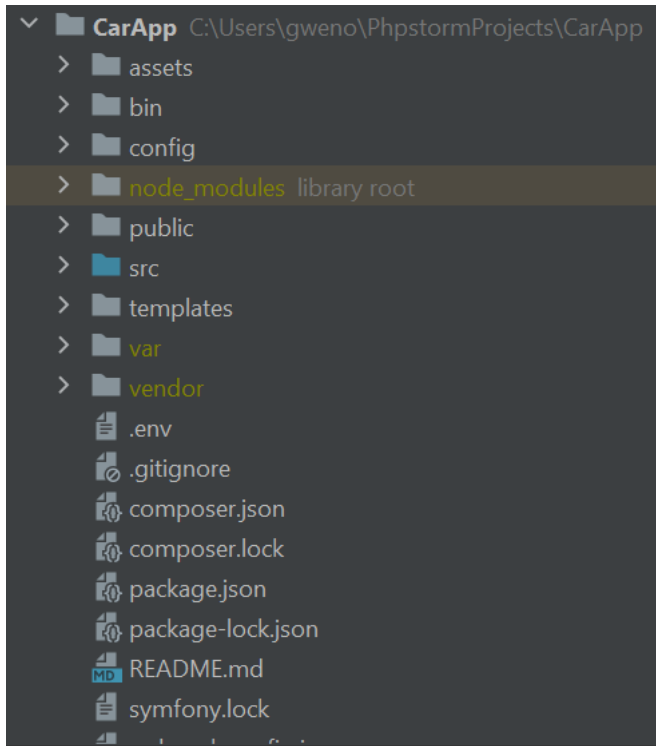
Cette page permet à l’administrateur d’accéder à quelques statistiques sur l’ensemble du site.

Cette page permet aussi à l’administrateur de gérer les rôles des utilisateurs : il peut changer le rôle entre CLIENT et LOUEUR. Il peut aussi supprimer un utilisateur.

## IV – Architecture et structure du code

Pour ce projet, nous avons utilisé Symfony et donc structuré notre projet en MVC (Model View Controller).

Voici les dossiers présents à la racine de notre projet.

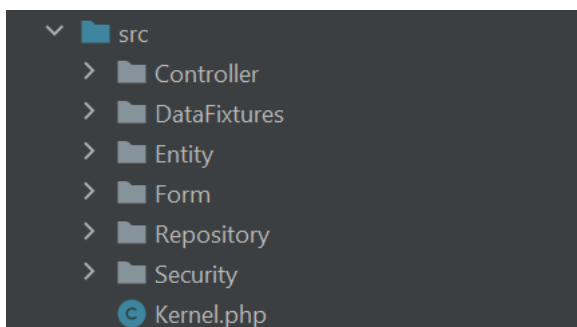


Le dossier assets contient la partie front-end avec les fichiers CSS et JavaScript. Pour intégrer ces fichiers, nous utilisons les appEntry intégrés à Symfony.

Le dossier config contient tous les fichiers permettant de configurer Symfony, comme doctrine.yaml ou security.yaml.

Le dossier public contient quelques images et une police, dans le sous-dossier media.

Le dossier src contient tous les fichiers qui permettent à l'application de fonctionner :



Le dossier Controller contient les controller qui vont récupérer les requêtes et appeler les templates twig ensuite.

Le dossier DataFixtures contient le fichier qui permet de charger des données dans la base lors de l'initialisation.

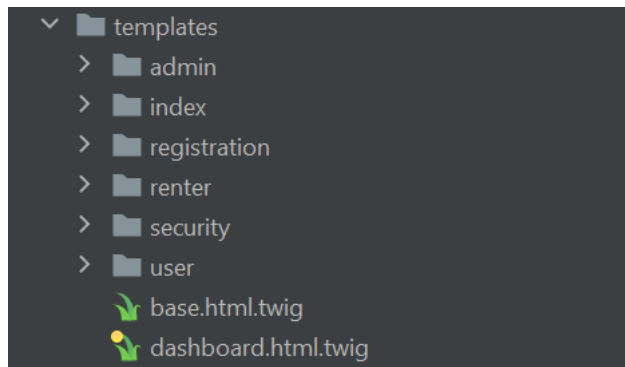
Le dossier Entity contient les entités utilisées pour ce projet : cela correspond aux tables de la base de données.

## Rapport de projet PWEB – Car’App

Le dossier Form contient le code qui permet de générer des formulaires, ce qui nous permet de faire des saisies robustes (ce code est appelé par les controller avant le render dans les templates twig). L’avantage est que ces codes peuvent être utilisés dans plusieurs controller.

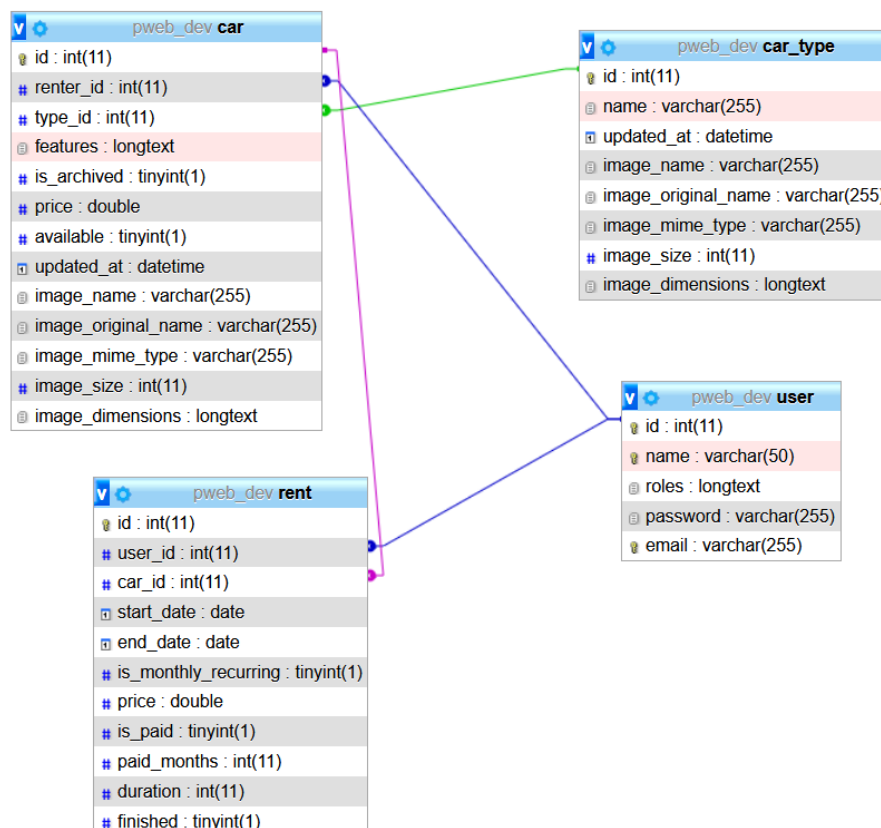
Le dossier Repository contient tous les modèles qui permettent de faire des requêtes et d’obtenir des données de la base. Nous utilisons doctrine et du SQL ici.

Le dossier Security contient le code permettant de gérer l’authentification sécurisée (ce qui inclut le stockage de mot de passe avec un rehashage régulièrement).



Le dossier templates contient toutes les templates appelées par les controller. Ces templates sont codées en twig et héritent entre elles. Par exemple, toutes les templates héritent de base.html.twig.

## V – Schéma relationnel de la base et explications



Voici le schéma de notre base.



# Rapport de projet

## PWEB – Car’App

Nous allons décrire rapidement chaque table.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 <b>id</b> 🔑	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer  Plus
<input type="checkbox"/>	2 <b>renter_id</b> 🔑	int(11)			Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	3 <b>type_id</b> 🔑	int(11)			Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	4 <b>features</b>	longtext	utf8mb4_unicode_ci		Oui	NULL	(DC2Type.json)		Modifier  Supprimer  Plus
<input type="checkbox"/>	5 <b>is_archived</b>	tinyint(1)			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	6 <b>price</b>	double			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	7 <b>available</b>	tinyint(1)			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	8 <b>updated_at</b>	datetime			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	9 <b>image_name</b>	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	10 <b>image_original_name</b>	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	11 <b>image_mime_type</b>	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	12 <b>image_size</b>	int(11)			Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	13 <b>image_dimensions</b>	longtext	utf8mb4_unicode_ci		Oui	NULL	(DC2Type.simple_array)		Modifier  Supprimer  Plus

### table car (véhicule)

Les 3 premiers champs contiennent respectivement l’id de la table *car*, l’id du loueur, et l’id du type de véhicule associé.

Le champ *features* contient les caractéristiques du véhicule.

Le champ *is\_archived* permet de savoir si la voiture a été supprimée par le loueur. Vu que les locations sont archivées, il ne fallait pas supprimer les véhicules pour pouvoir conserver toutes les informations.

Le champ *price* correspond au prix du véhicule par jour, en €.

Le champ *available* indique l’état du véhicule : 0 indisponible et 1 disponible.

Les champs 8 à 13 correspondent à des valeurs nécessaires au plugin qui permet l’upload des images (VichUploadBundle).

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 <b>id</b> 🔑	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer  Plus
<input type="checkbox"/>	2 <b>name</b>	varchar(255)	utf8mb4_unicode_ci		Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	3 <b>updated_at</b>	datetime			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	4 <b>image_name</b>	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	5 <b>image_original_name</b>	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	6 <b>image_mime_type</b>	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	7 <b>image_size</b>	int(11)			Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	8 <b>image_dimensions</b>	longtext	utf8mb4_unicode_ci		Oui	NULL	(DC2Type.simple_array)		Modifier  Supprimer  Plus

### table car\_type (type de véhicule)

Le premier champ contient l’id du type.

Le champ *name* correspond au nom du type (« Audi » par exemple).

Les champs 3 à 8 correspondent à des valeurs nécessaires au plugin qui permet l’upload des images (VichUploadBundle).

## Rapport de projet PWEB – Car’App

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 <b>id</b> 🔑	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer  Plus
<input type="checkbox"/>	2 <b>user_id</b> 🔑	int(11)			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	3 <b>car_id</b> 🔑	int(11)			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	4 <b>start_date</b>	date			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	5 <b>end_date</b>	date			Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	6 <b>is_monthly_recurring</b>	tinyint(1)			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	7 <b>price</b>	double			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	8 <b>is_paid</b>	tinyint(1)			Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	9 <b>paid_months</b>	int(11)			Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	10 <b>duration</b>	int(11)			Oui	NULL			Modifier  Supprimer  Plus
<input type="checkbox"/>	11 <b>finished</b>	tinyint(1)			Oui	NULL			Modifier  Supprimer  Plus

*table rent (location)*

Les 3 premiers champs correspondent respectivement à l’id de la location, à l’id de l’utilisateur qui loue la voiture et à l’id du véhicule.

Les champs start\_date et end\_date correspondent aux dates de début et de fin de la location. Si la location est « mensuelle », le champ end\_date contient NULL.

Le champ is\_monthly\_recurring permet de savoir si la location est mensuelle ou non (booléen).

Le champ price correspond au prix total d’une location, ou au prix mensuel si la location est mensuelle.

Le champ is\_paid permet de savoir si la location a été payée. Si la location est mensuelle, ce champ est à 1 pendant un mois quand le paiement est effectué.

Le champ paid\_months permet de savoir combien de mois ont déjà été payés par le client pour savoir combien de jours ou de mois il doit payer, dans le cas d’une location mensuelle.

Le champ duration permet de savoir combien il reste de jours à la location. Il contient NULL si c’est une location mensuelle.

Le champ finished permet de savoir si la location est « terminée » ou non, pour l’archivage. Une location « terminée » est une location qui a été payée et qui n’est pas mensuelle ou qui a été arrêtée par le client.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 <b>id</b> 🔑	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer  Plus
<input type="checkbox"/>	2 <b>name</b> 🔑	varchar(50)	utf8mb4_unicode_ci		Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	3 <b>roles</b>	longtext	utf8mb4_unicode_ci		Non	Aucun(e)	(DC2Type:json)		Modifier  Supprimer  Plus
<input type="checkbox"/>	4 <b>password</b>	varchar(255)	utf8mb4_unicode_ci		Non	Aucun(e)			Modifier  Supprimer  Plus
<input type="checkbox"/>	5 <b>email</b> 🔑	varchar(255)	utf8mb4_unicode_ci		Non	Aucun(e)			Modifier  Supprimer  Plus

*table user (utilisateur)*

Le champ id correspond à l’id de l’utilisateur de l’application.

Le champ nom contient le nom de l’utilisateur, ce champ est marqué UNIQUE pour respecter les bonnes pratiques de sécurité Symfony.

## Rapport de projet PWEB – Car’App

Le champ rôle contient son rôle. Les 3 rôles possibles sont `ROLE_ADMIN`, `ROLE_LOUEUR` et `ROLE_CLIENT`, écrits en JSON dans le champ.

Le champ password contient le mot de passe de l'utilisateur, celui-ci est stocké de façon crypté et réhashé régulièrement par l'application.

Le champ email contient l'adresse mail de l'utilisateur, elle doit être unique (pas deux utilisateurs avec la même adresse mail sur le site).

## VI – Outils

Nous avons utilisé de nombreux outils pour ce projet :

- PhpStorm
- Git pour coder ensemble et merge nos codes
- Framework Symfony
- Node.js
- Composer
- Yarn
- phpMyAdmin, pour la gestion rapide de la base de données

Nous avons aussi utilisés ces extensions dans notre code (liste non exhaustive) :

### Back-end

- Twig : gérer les vues du site grâce à des templates avancées
- Doctrine ORM : gérer les modèles, les entités et la connection à la base de données
- Webpack Encore : gérer tous les assets front-end, injecte le css, le javascript et les librairies supplémentaires
- Symfony/form : gérer la création et le traitement des formulaires
- Symfony/maker-bundle : créer des classes types rapidement et efficacement
- Symfony/security-bundle : gérer les connections et les rôles de chaque utilisateurs
- Symfony/validator : vérifie les données des formulaires pour correspondre au schéma de la base de données
- Symfony/web-profiler-bundle : barre de debug en environnement de développement

### Front-end

- Chart.js : affichage de graphique dynamique
- bsCustomFileInput : importation de fichier avec bootstrap

Nous avons aussi utilisé Twig pour les templates.

## VII – Bilan du projet

Nous avons beaucoup apprécié faire ce projet. Il nous a permis de découvrir le framework Symfony, très utile pour le développement de ce type d’application.

Après la prise en main de ce framework, nous avons pu développer de façon fluide, en utilisant Git pour fusionner nos codes.

Avec l’architecture utilisée pour ce projet, il est facile d’y faire des extensions ou de continuer à développer des services. Un projet est toujours perfectible mais nous pensons avoir répondu aux attentes de celui-ci.

Notre guide d’installation, disponible dans le readme, est très complet et peut permettre à un développeur de continuer le projet s’il en a envie. C’est un projet que nous serons tous fiers d’afficher sur nos CV car il constitue l’un des projets les plus complets que nous avons réalisé pendant notre DUT.