

# RAPPORT PROJET

**ASP.NET Core MVC**  
*Car rental plateforme*

**JANVIER 8**

**Authored by:**

**CHAKIB Amine**

**EL YOUSSEFI ALAOUI Abdellah**

**MTOUAA Mourad**



**Scan this QR Code to get to the  
source code**

---

# Table de matières

Chapitre I : Vision globale du projet .....	4
Cadre du travail .....	4
Présentation du projet .....	4
Approches de travail : .....	4
Cahier des charges .....	5
Les Spécifications du Projet .....	6
Charte Graphique.....	9
Contraintes : .....	9
Chapitre II – Conception et modélisation .....	11
Use case scenario .....	11
Workflow – Microsoft Visio - .....	12
Architecture du base de données .....	13
Chapitre III : Réalisation.....	14
Technologies utilisées .....	14
Gestion de code source décentralisé .....	14
Déploiement de la base de données.....	15
Front End .....	15
Pagination.....	15
Support multilingue .....	15
Utilisations des APIs .....	15
Quelques captures d’écran de la plateforme.....	16

---

# Liste des figures

Tableau 1 : Espace Propriétaire .....	6
Tableau 2 : Gestion des utilisateurs .....	6
Tableau 3 : Création de compte .....	7
Tableau 4 : Authentification .....	7
Tableau 5 : Espace Locataire .....	7
Figure 1: Use case scenario .....	11
Figure 2: Workflow - Microsoft Visio -.....	12
Figure 3 : Architecture de la base de données .....	13
Figure 4 : Page d'authentification .....	16
Figure 5 : Page de création du compte .....	16
Figure 6 : Espace propriétaire (Espace Voiture) .....	17
Figure 7 : Liste de demandes de locations .....	17
Figure 8 : Ajout d'un utilisateur par un administrateur .....	18
Figure 9 : Gestion des utilisateurs .....	18
Figure 10 : Liste noire.....	19
Figure 11 : Gestion des modèles .....	19
Figure 12 : Gestion des marques .....	20
Figure 13: Espace Locataire (Francais) .....	20
Figure 14: Espace Locataire (English).....	21
Figure 15 : Système de pagination .....	21
Figure 16 : Gestion de favoris.....	22

---

# Chapitre I : Vision globale du projet

## Cadre du travail

### **Présentation du projet**

Notre mission dans le cadre de ce projet est de concevoir et développer une plateforme d'allocation des voitures P2P. Il s'agit d'une application web destinée aux propriétaires de voitures soit agences ou individus, locataires et administrateurs relatifs à la plateforme. L'idée est de définir les fonctionnalités suivantes, que doit l'application offrir aux utilisateurs : Locations de voitures, Espace locataire, espace propriétaire, gestion du marques et modèles disponibles, gestion du voitures, gestion de la liste d'offre de location (Favoris, Allouer...), Gestion d'utilisateurs...

### **Approches de travail :**

Le projet comprend trois phases : la première est la phase fonctionnelle, la deuxième est la phase technique et la troisième est la phase de réalisation.

#### ***a. Phase fonctionnelle :***

Elle vise la capture des besoins fonctionnels et leur analyse de manière à déterminer ce que va réaliser le système. C'est ici, qu'on identifie et dégage toutes les fonctionnalités du projet à réaliser.

#### ***b. Phase Technique :***

Capitalise un savoir-faire technique et permet la capture des besoins non fonctionnels. Elle constitue aussi un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent être indépendantes des fonctions à réaliser.

#### ***c. Phase Applicative :***

Cette phase consiste à fusionner les résultats des deux phases fonctionnelle et technique. Cette fusion mène à la conception applicative et à la solution adaptée aux besoins des utilisateurs.

# Cahier des charges

## Identification des acteurs

<i>Acteurs</i>	<i>Fonctions</i>
Propriétaire	<ul style="list-style-type: none"><li>• Gestion de voitures :<ul style="list-style-type: none"><li>▸ Ajouter une voiture.</li><li>▸ Modifier une voiture.</li><li>▸ Supprimer une voiture.</li><li>▸ Accepter ou refusé les demandes de location.</li></ul></li></ul>
Locataire	<ul style="list-style-type: none"><li>▸ Consulter la liste des voitures disponibles et louées.</li><li>▸ Effectuer des recherches par mot clé ou par critère.</li><li>▸ Envoyer des demandes de location aux propriétaires des voitures.</li><li>▸ Ajouter une voiture à la liste des favoris.</li><li>▸ Consulter les détails d'une voiture.</li></ul>
Administrateur	<ul style="list-style-type: none"><li>• Gestion de voitures :<ul style="list-style-type: none"><li>▸ Ajouter une voiture.</li><li>▸ Modifier une voiture.</li><li>▸ Supprimer une voiture.</li></ul></li><li>• Gestion de utilisateurs :<ul style="list-style-type: none"><li>▸ Ajouter un utilisateur et contribuer le rôle Admin ou Exécutive.</li><li>▸ Modifier un utilisateur.</li><li>▸ Supprimer un utilisateur.</li><li>▸ Ajouter un utilisateur à la liste noire.</li><li>▸ Retirer un utilisateur de la liste noire.</li></ul></li><li>• Gestion des Marques de voiture :<ul style="list-style-type: none"><li>▸ Ajouter une marque.</li><li>▸ Modifier une marque.</li><li>▸ Supprimer une marque.</li></ul></li><li>• Gestion des Modèles de voiture :<ul style="list-style-type: none"><li>▸ Ajouter un Modèle.</li><li>▸ Modifier un Modèle.</li><li>▸ Supprimer un Modèle.</li></ul></li></ul>
Visiteur	<ul style="list-style-type: none"><li>▸ Afficher la liste des voitures disponibles et louées.</li><li>▸ Effectuer des recherches par mot clé ou par critère.</li><li>▸ Afficher les détails d'une voiture.</li><li>▸ Créer un compte.</li><li>▸ S'authentifier.</li></ul>

## Les Spécifications du Projet

### **Analyse des besoins :**

L'analyse du besoin consiste à définir de manière précise et exhaustive le besoin, C'est-à-dire ce qui est exprimé ou non exprimé par l'utilisateur et qui lui permet de réaliser un certain nombre d'actions ou tâches.

Cette étude consistera à mettre en œuvre un certain nombre d'outils d'analyse qui permettront de modéliser le besoin et de le rendre compréhensible par ceux qui vont le couvrir.

### **Exigences fonctionnelles :**

Cette étude consiste à définir les fonctions que le produit ou service devra remplir. Chacune des fonctions doit être qualifiée par un niveau de performance. Les fonctions seront hiérarchisées entre elles afin de préparer d'éventuels arbitrages si on ne peut pas techniquement ou financièrement réaliser toutes les fonctions.

Fonction : Gestion des voitures	
<b>Objectif</b>	<ul style="list-style-type: none"><li>▶ Gérer les voitures des propriétaires.</li><li>▶ Gérer les demandes de location effectuer par les locataires.</li></ul>
<b>Description</b>	<ul style="list-style-type: none"><li>▶ Un propriétaire peut consulter la liste de ses voitures disponible et occuper.</li><li>▶ Un locataire peut accepter ou refusé les demandes de location effectuer par les locataires, s'il accepte alors la voiture passe de l'état disponible à l'état occuper.</li><li>▶ Un propriétaire peut ajouter une ou plusieurs voitures à la liste des voitures.</li></ul>
<b>Contrainte</b>	<ul style="list-style-type: none"><li>▶ Le propriétaire doit être authentifier.</li><li>▶ Le propriétaire ne doit pas être dans la liste noire.</li></ul>

*Tableau 1 : Espace Propriétaire*

Fonction : Gestion des utilisateurs	
<b>Objectif</b>	<ul style="list-style-type: none"><li>▶ Gérer les utilisateurs du compte (ajout d'utilisateurs, attributions d'autorisations ).</li><li>▶ Accorder ou retirer les droits d'accès à tous les utilisateurs sauf les Admins.</li></ul>
<b>Description</b>	<ul style="list-style-type: none"><li>▶ Un Admin peut ajouter d'autres Admin ou utilisateurs normales ; Ainsi que Admin à le pouvoir de retirer les droits d'accès à un compte en ajoutant se dernier à la liste noire comme il peut aussi retirer un utilisateur de la liste noire.</li><li>▶ Un utilisateur authentifier peut accéder à ses ressources, envoyer une demande de location d'une ou plusieurs voitures</li><li>▶ Un utilisateur peut ajouter des voitures à la liste des favoris</li><li>▶ Un propriétaire peut soi accepter soi refuser la demande de location d'un locataire.</li></ul>

<b>Contraintes</b>	<ul style="list-style-type: none"> <li>▶ L'Admin doit être authentifier.</li> <li>▶ L'Admin et le seul à pouvoir ajouter d'autres Admin ou utilisateurs normales ;</li> </ul>
--------------------	---

*Tableau 2 : Gestion des utilisateurs*

<b>Fonction : Création de compte</b>	
<b>Objectif</b>	S'inscrire pour pouvoir exploiter l'application.
<b>Description</b>	- <b>La page de création possède les champs suivants</b> : Username/Nom, Prénom, Numéro de téléphone, Ville, Adresse e-mail, Mot de passe, Confirmer mot de passe , Button « <i>Créer un compte</i> » qui permet la création du compte
<b>Contraintes</b>	<ul style="list-style-type: none"> <li>▶ Un utilisateur déjà authentifier ne peut pas y accéder.</li> <li>▶ Un nom d'utilisateur déjà existant et la saisie des informations qui ne respecte pas les normes(contrainte) redirige vers la page de création.</li> </ul>

*Tableau 3 : Création de compte*

<b>Fonction : Authentification</b>	
<b>Objectif</b>	Chaque utilisateur possède un nom d'utilisateur et un mot de passe spécifique qui lui permet de vérifier son identité, afin d'autoriser l'accès de cette entité à des ressources en toute sécurité.
<b>Description</b>	- <b>La page d'authentification possède les champs suivants</b> : Username, Mot de passe, Button « <i>Login</i> » et un bouton Register qui permet une redirection vers le formulaire d'inscription si l'utilisateur n'as pas encore de compte.
<b>Contraintes</b>	<ul style="list-style-type: none"> <li>▶ Un utilisateur déjà authentifier ne peut pas y accéder.</li> <li>▶ Un nom d'utilisateur ou mot de passe incorrecte redirige vers la page d'authentification.</li> </ul>

*Tableau 4 : Authentification*

<b>Fonction : Gestion des locations</b>	
<b>Objectif</b>	Gérer les locations d'un Utilisateur.
<b>Description</b>	<ul style="list-style-type: none"> <li>▶ Un locataire peut consulter la liste des voitures disponible et occuper.</li> <li>▶ Un locataire peut envoyer une demande de location au propriétaire qui à le choix d'accepter ou refusé la demande, s'il accepte alors la voiture passe de l'état disponible à l'état occuper.</li> <li>▶ Un locataire peut effectuer une recherche de voiture soit par mot clé soit par critère (Marque, Couleur, Prix, Disponibilité).</li> <li>▶ Un locataire peut ajouter une ou plusieurs voitures à la liste des favoris.</li> <li>▶ Un utilisateur peut ajouter des voitures à la liste des favoris</li> <li>▶ Un locataire peut consulter l'historique des voitures louer.</li> </ul>
<b>Contraintes</b>	<ul style="list-style-type: none"> <li>▶ Le Locataire doit être authentifier.</li> <li>▶ Le Locataire ne doit pas être dans la liste noire.</li> </ul>

*Tableau 5: Gestion des locations*

---

### **Contenu de la base de données :**

- **Informations sur les utilisateurs :** Nom, prénom, email, adresse, Numéro Téléphone, Rôle (*Admin, Executive, Blacklisted*)...
- **Liste des marques.**
- **Liste des modèles.**
- **Informations sur les voitures :** Marque, Model, Couleur, Année, Kilométrage, Prix de location par jour, Image, Propriétaire, Description...
- **Informations sur les locations :** Durée, Etat de location, Prix total...
- **Liste de demande de locations.**
- **Liste de favoris.**

### **Exigences techniques :**

Elles consistent à choisir ou concevoir des solutions techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et en ce qui concerne notre application web, nous avons dégagé l'ensemble des besoins suivants :

- **Performance:**
  - Temps de réponse (*Utilisation du système de pagination pour minimiser la charge de données*)
  - Tolérance aux pannes.
- **Disponibilité :** l'application doit être disponible pour être utilisée par n'importe quel utilisateur par n'importe quel appareil (*Responsive en utilisant le system Grid*).
- **Sécurité :** l'application contrôlera chaque point d'entrée potentiel d'utilisateurs malveillants, le code de l'application joue évidemment un rôle clef au niveau de chiffrement des données sensibles :
  - Utilisation du *ASP.NET Core Identity* chiffre les mots de passes ;
  - Gestion d'accès pour chaque interface ;
- **Maintenabilité :** l'application doit être évolutive dans le temps, fermée à la modification ouverte à l'extension (*Utilisation d'injection dépendance*).
- **Portabilité :** L'application doit être portable et déployable dans différents environnement (*Utilisation du ASP.NET Core*).



---


## Charte Graphique

### *Choix des couleurs*

On avait la liberté de choisir les couleurs, c'est pour cela on a décidé d'utiliser dans l'intégralité de l'application la couleur **Teal** (**#008080**).

**Teal** combine les propriétés calmantes du bleu avec les qualités de renouvellement du vert. C'est une couleur revitalisante et rajeunissante qui représente également une communication ouverte et une clarté de pensée. Pour les moines tibétains, la sarcelle d'hiver symbolise l'infinité de la mer et du ciel, alors qu'elle est la couleur de la vérité et de la foi pour les Égyptiens.

### *Choix d'identité visuelle*

 Un logotype est une représentation graphique d'une marque ou d'une entreprise, il est le résultat d'une recherche créative approfondie visant à représenter et valoriser l'image de notre projet en le donnant une personnalité morale. Un logotype se caractérise par le choix de ses couleurs, le choix d'une typographie et souvent celui d'un symbole.

## Contraintes :

### *Contraintes en termes de délais :*

À partir de la livraison du cahier des charges, nous disposons d'environ **trois semaines** pour la présentation du livrable au sein du ENSA comme un examen final pour l'élément du module asp.net. Le délai semble très court pour le projet entier dédié à la réalisation de toutes les fonctionnalités, mais reste suffisant pour se concentrer sur la partie prévue.

### *Contraintes de sécurité :*

La gestion de sécurité est la principale contrainte de notre système. L'application doit posséder une gestion de privilèges et de niveaux d'accès pour les différents types d'utilisateurs (Administrateur, Propriétaire, Locataire). Selon leur données personnelles, le contenu des pages varie et l'accès aux informations avec un autre utilisateur est interdit car la base de données contient des informations personnelles

---

comme l'historique des locations, les mots de passe, ainsi que le droit de modification de données personnels et de voitures...

***Contraintes techniques :***

Pour le développement de notre système, nous devons baser notre application sur la flexibilité de différents systèmes. La structure de notre système doit être extensible pour la mettre en place facilement. De plus, le développement devra suivre toutes les normes techniques pour une meilleure performance, maintenance et facilité de mise à jour.

En plus, dans le cahier de charge la technologie utilisée était explicitement indiquée par une simple raison que c'est un projet du module **.NET Technologies**.

Pour notre intension d'extension et d'ouvrir le code pour le monde Open Source ainsi que multiplateforme, l'utilisation du **ASP.NET Core** est essentiel.

# Chapitre II – Conception et modélisation

## Use case scenario

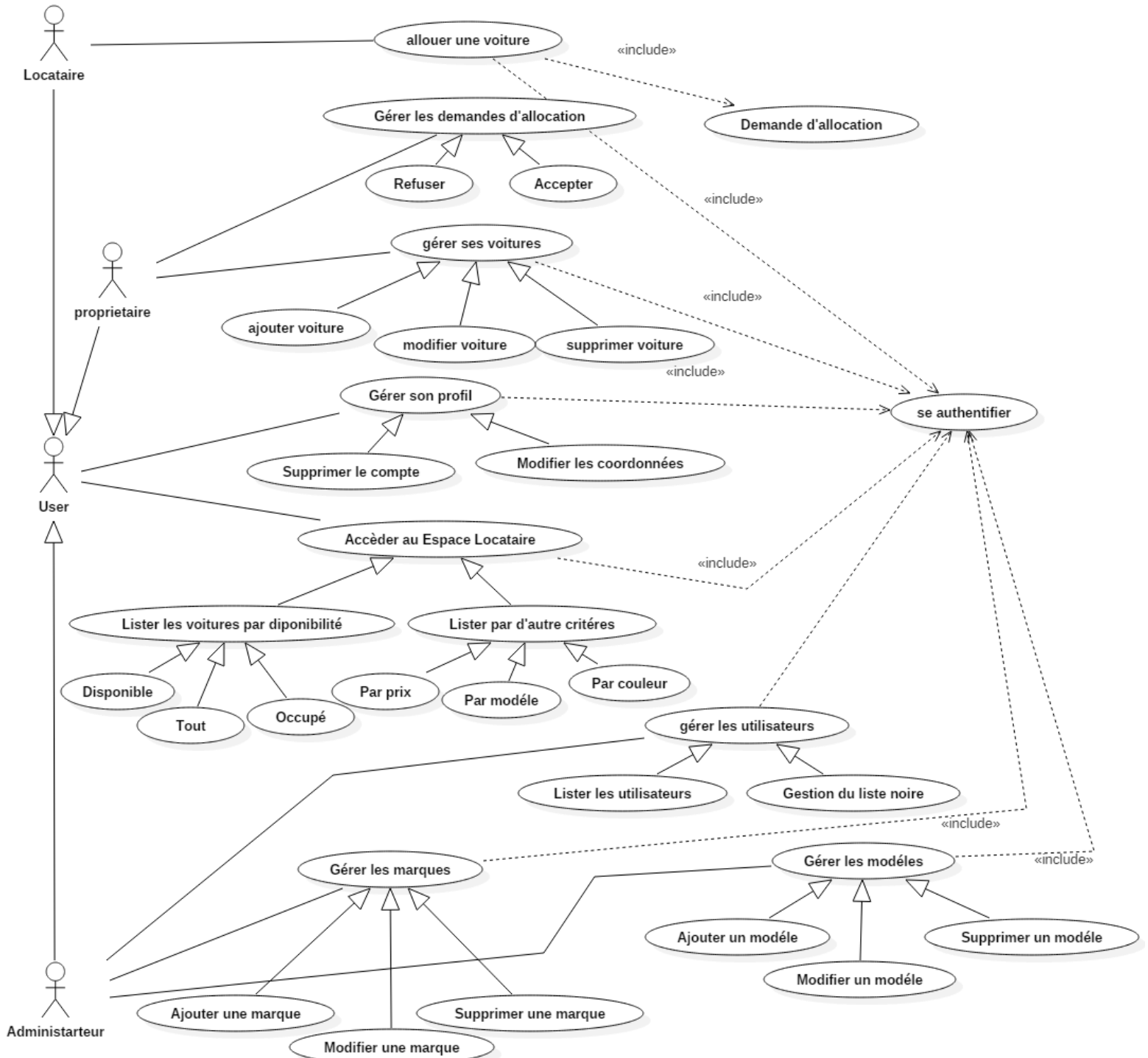


Figure 1: Use case scenario

## Workflow – Microsoft Visio -

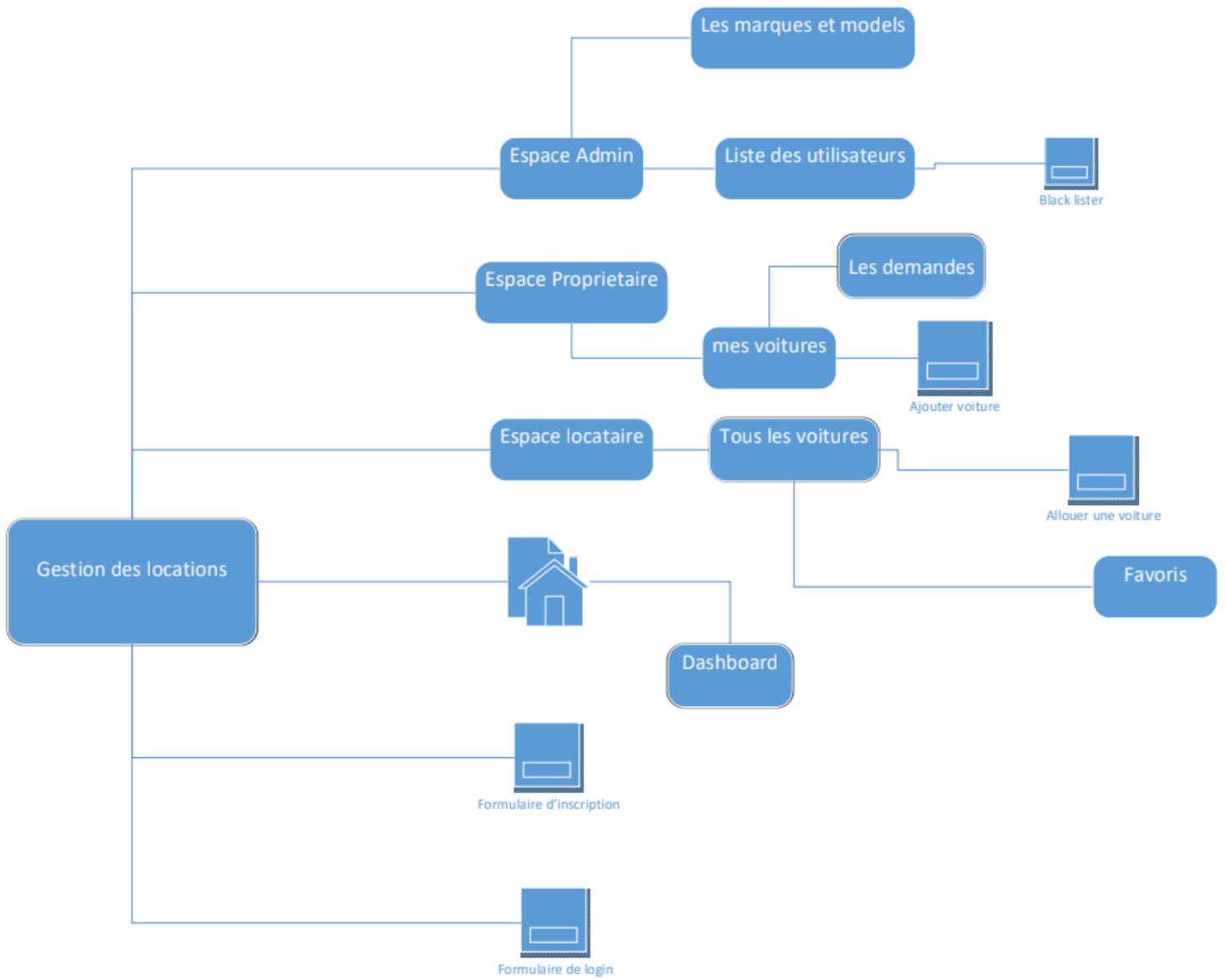


Figure 2: Workflow - Microsoft Visio -

## Architecture du base de données

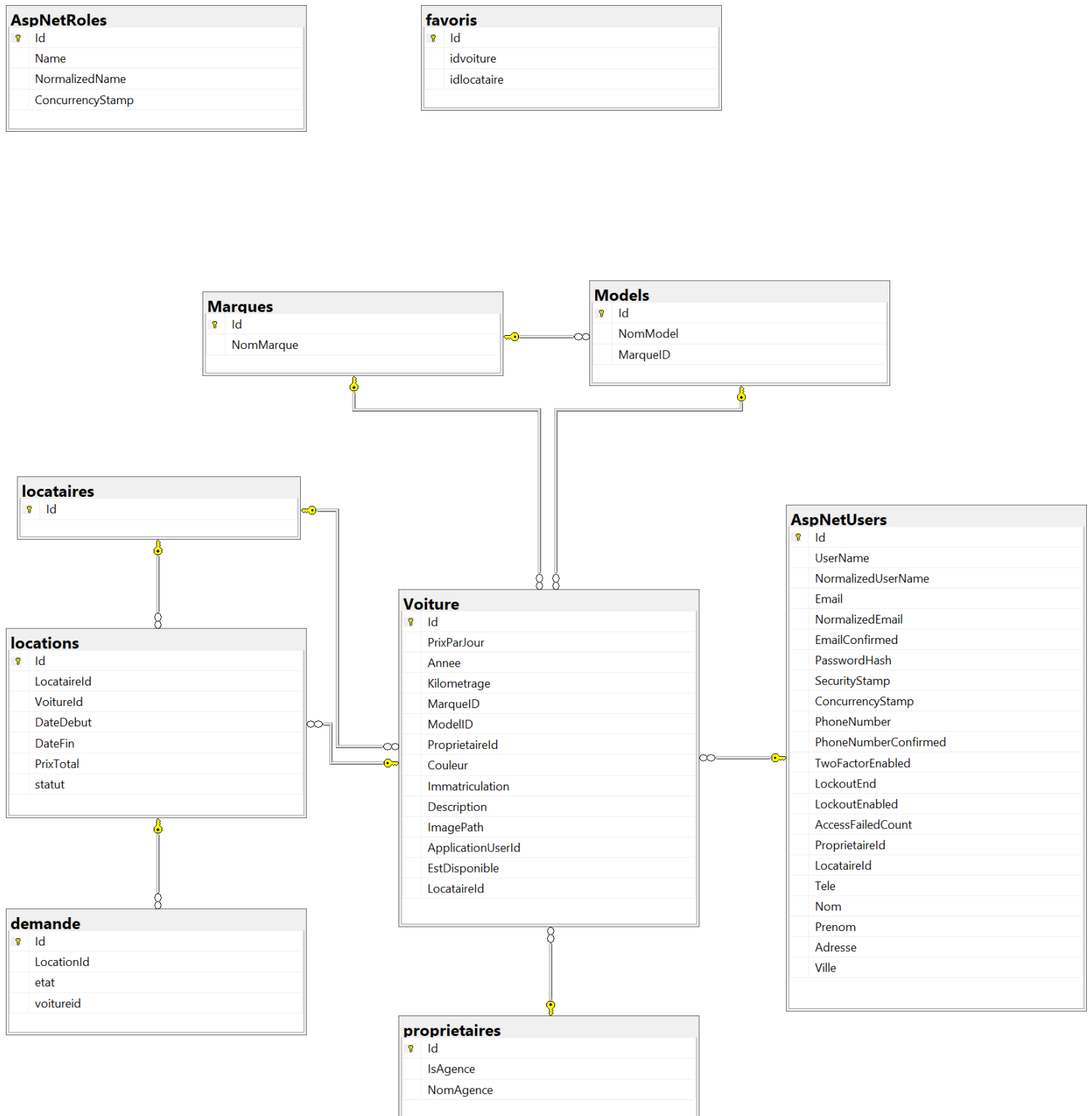


Figure 3 : Architecture de la base de données

# Chapitre III : Réalisation

## Technologies utilisées

 <b>Azure</b>	<b>Microsoft Azure</b> est un service de <i>cloud computing</i> créé par Microsoft pour créer, tester, déployer et gérer des applications et des services via des centres de données gérés par Microsoft.
<b>Google Chart API</b>	L' <b>API Google Chart</b> est un service Web interactif qui crée des graphiques à partir de données fournies par l'utilisateur. Le service prend en charge une grande variété d'informations et de formatage de cartes.
	<b>Ajax</b> est un ensemble de techniques de développement Web utilisant de nombreuses technologies Web côté client pour créer des applications Web asynchrones. Avec Ajax, les applications Web peuvent envoyer et récupérer des données d'un serveur de manière asynchrone sans interférer avec l'affichage et le comportement de la page existante
	Le framework <b>MVC ASP.NET Core</b> est un framework de présentation léger, open source et hautement testable, optimisé pour une utilisation avec ASP.NET Core. ASP.NET Core MVC fournit un moyen basé sur des modèles pour créer des sites Web dynamiques qui permet une séparation nette des problèmes.
	<b>Semantic UI</b> est un cadre de développement frontal moderne (User interface), propulsé par LESS et jQuery. Il a un design élégant, subtil et plat qui offre une expérience utilisateur légère. Selon le site Web Semantic UI, l'objectif du cadre est de responsabiliser les concepteurs et les développeurs «En créant un langage pour partager l'interface utilisateur».
	<b>Git</b> est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes
<b>JWT</b>	<b>JSON Web Token</b> est une norme Internet pour la création de jetons d'accès basés sur JSON qui revendiquent un certain nombre de revendications. Par exemple, un serveur pourrait générer un jeton qui a la revendication « Connecté en tant qu'administrateur » et le fournir à un client. Le client pourrait alors utiliser ce jeton pour prouver qu'il est connecté en tant qu'administrateur.

### Gestion de code source décentralisé

Pour la distribution de tâches entre équipe, et pour avoir l'accès au code source dans n'importe quelle machine, et le poursuit des versions d'application, nous avons mis en place notre code source en chaque étape et le synchroniser avec le *repository* GitHub (<https://github.com/mouradxmt/carRental>)

---

## Déploiement de la base de données.

Pour que le produit soit disponible partout, et ainsi pour avoir une base de données centralisé ; la centralisation du code source n'était pas la seule solution nécessaire, mais nous étions obligés de déployer au fur et à mesure la base de données dans un serveur **Microsoft Azure**.

## Front End

Dans le front end, c'était très primordiale de tenir en compte UI/UX, L'interface utilisateur et ainsi l'expérience d'utilisateur dans toutes les interfaces.

Pour cela on a utilisé le Framework **Semantic UI**, qui prend en considération la sémantique du code HTML.

Suivant la **loi européenne**, tous les site-web utilisent les cookies doivent le déclarer, alors qu'on a pensé à cela, et un consentement d'existence des cookies s'affiche.



This website uses cookies to ensure you get the best experience on our website.

[Learn More](#)[Accept](#)

## Pagination

Pour une flexibilité d'utilisation, et pour une rapidité de chargement de page, le système de pagination était nécessaire de l'implémenter dans L'espace Locataire qui prend en charge plusieurs voitures qui sont listés pour locations.

## Support multilingue

Bien entendue, la plateforme doit être extensible, flexible, performante, multi plateforme, mais aussi doit être accessible par plusieurs personnes, non seulement les **francophones**, mais aussi les **anglophones** et d'autres...

Dans notre projet nous avons supporté le **multilingue**, et la plateforme dans toute son entité supporte la langue **Française** et **Anglaise**.

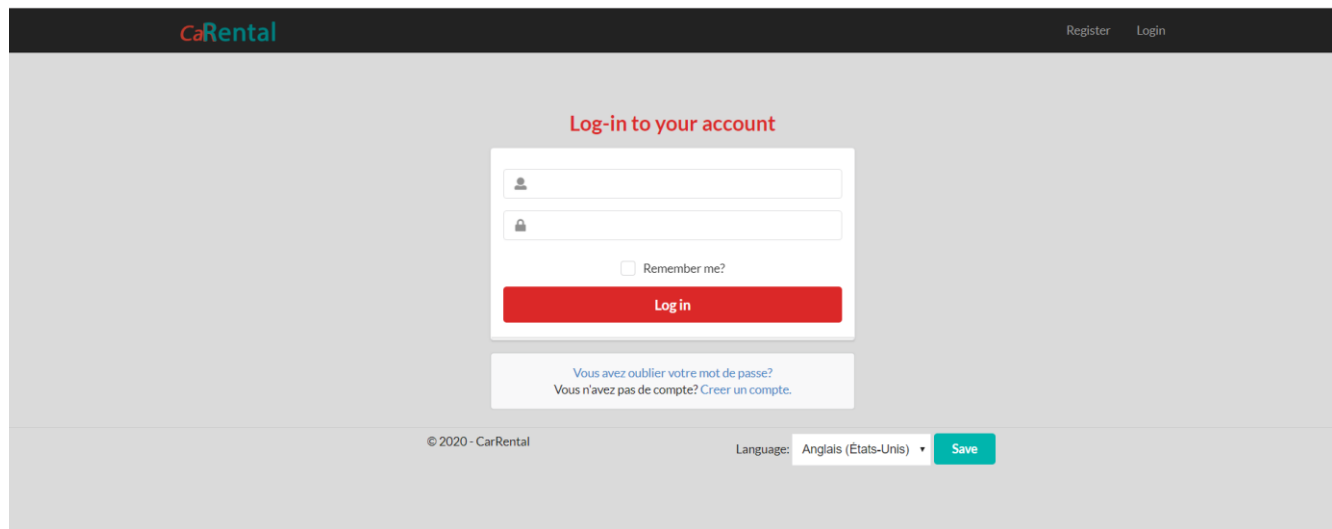
## Utilisations des APIs

API signifie *Application Programming Interface* permet d'effectuer des traitements et envoyer les données au format JSON ou XML et envoyer ses données à un autre service pour les traiter.

La plateforme ne restera pas toujours la seule consommatrice de ses services, on peut créer par la suite une application Android/iOS, ou une application Desktop, Donc créer des APIs qui seront utiles par la suite.

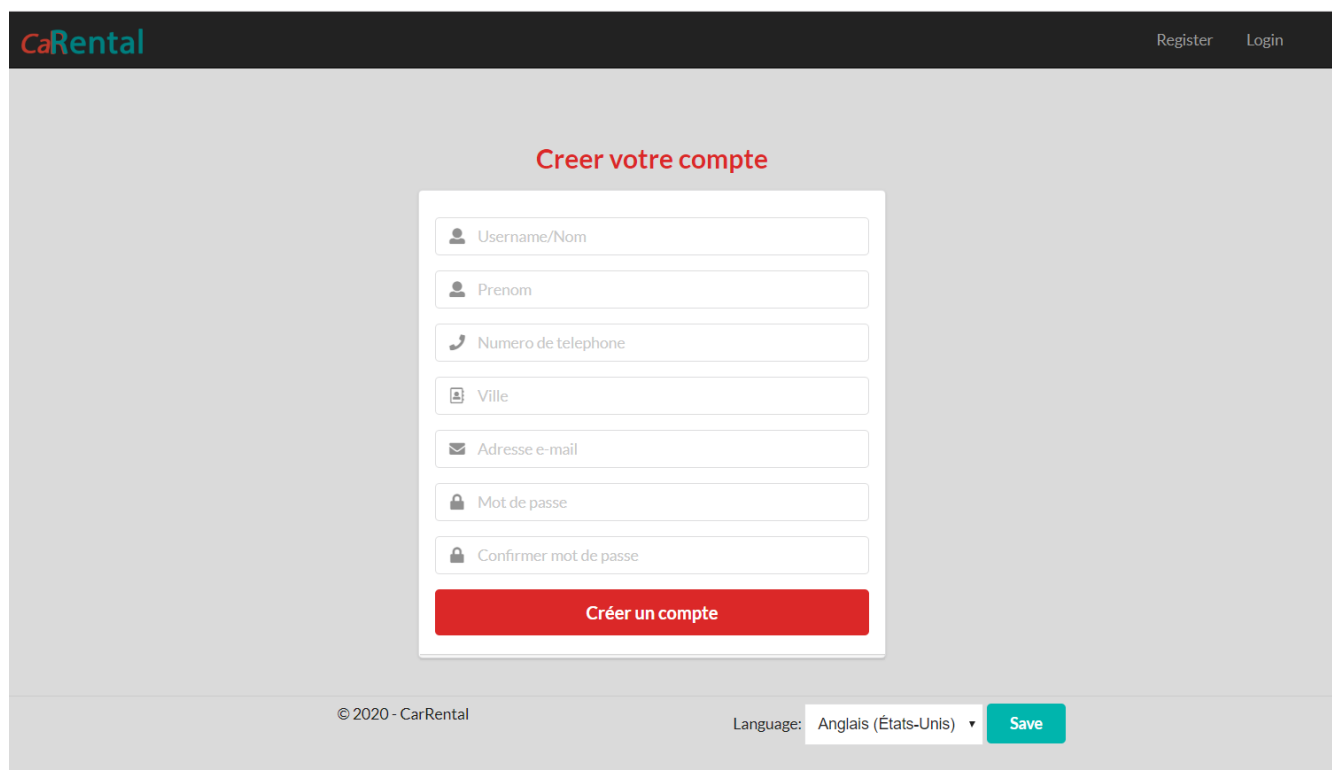
Pour sa sécurité on utilisera **JWT**.

## Quelques captures d'écran de la plateforme



The screenshot shows the login page of the CaRental platform. At the top, there is a dark header with the CaRental logo on the left and 'Register' and 'Login' links on the right. The main content area has a light gray background. In the center, there is a white box titled 'Log-in to your account'. Inside this box, there are two input fields: the first for a username (with a person icon) and the second for a password (with a lock icon). Below these fields is a checkbox labeled 'Remember me?'. A red 'Log in' button is positioned below the checkbox. Underneath the login box, there is a link that says 'Vous avez oublié votre mot de passe? Créez un compte.' At the bottom of the page, there is a footer with '© 2020 - CarRental' on the left, a language selector showing 'Anglais (États-Unis)' in the middle, and a green 'Save' button on the right.

**Figure 4 : Page d'authentification**



The screenshot shows the account creation page of the CaRental platform. The header is identical to the login page. The main content area features a white box titled 'Créer votre compte'. This box contains seven input fields, each with a specific icon: 'Username/Nom' (person), 'Prenom' (person), 'Numero de telephone' (phone), 'Ville' (location pin), 'Adresse e-mail' (envelope), 'Mot de passe' (lock), and 'Confirmer mot de passe' (lock). A red 'Créer un compte' button is located at the bottom of the form. The footer is also identical to the login page, showing '© 2020 - CarRental', the language selector, and the 'Save' button.

**Figure 5 : Page de création du compte**



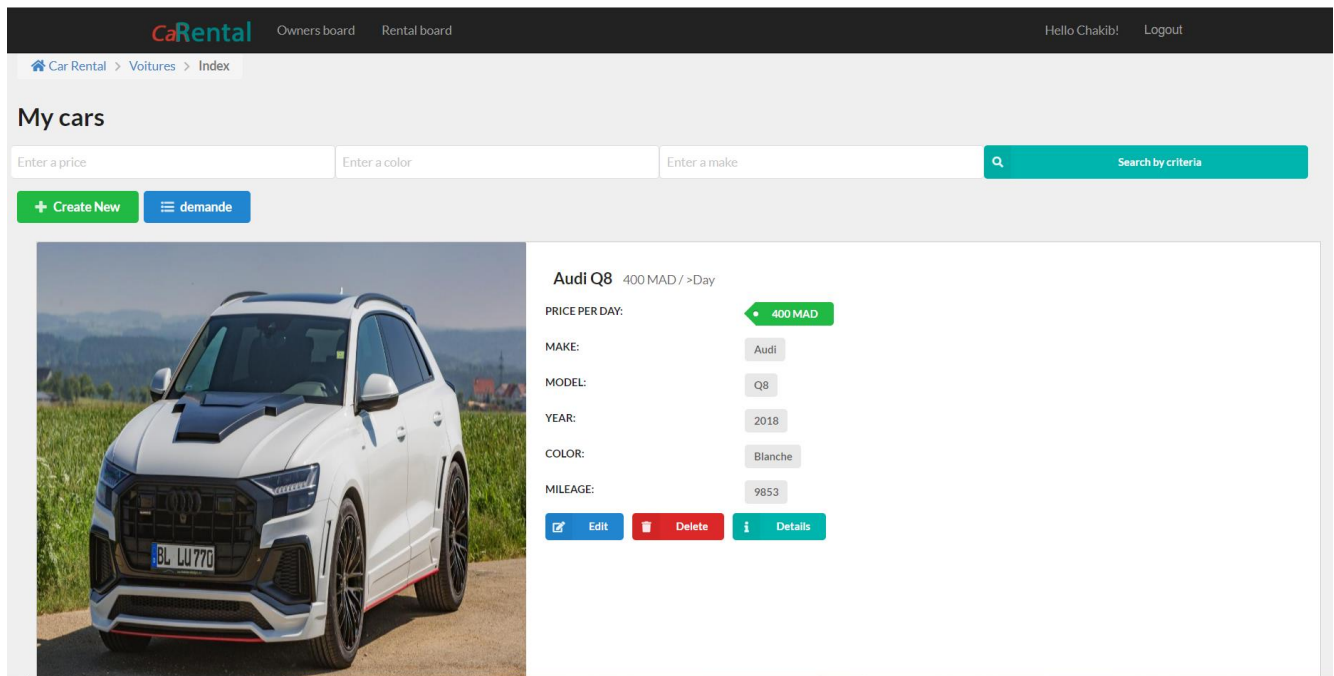


Figure 6 : Espace propriétaire (Espace Voiture)

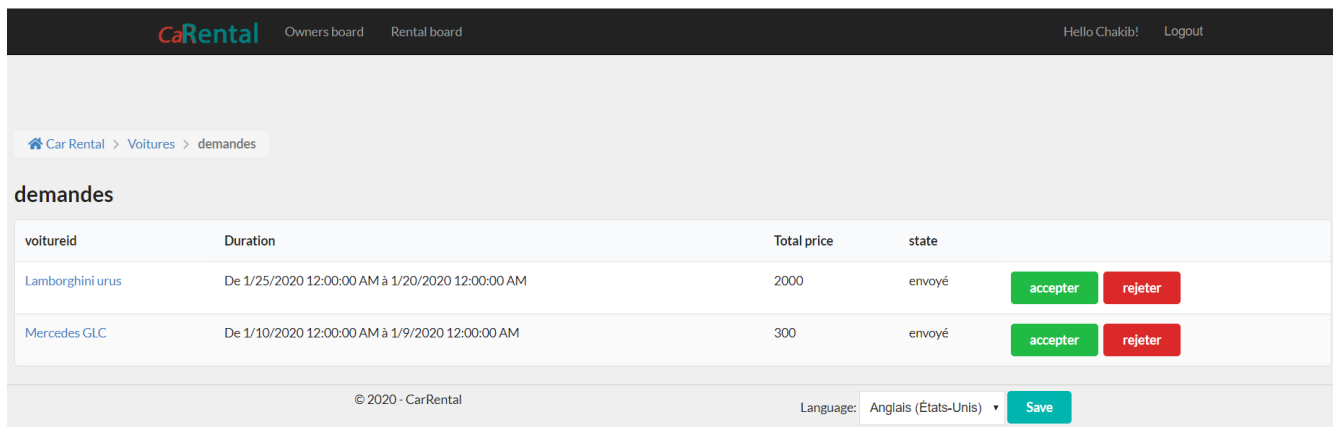


Figure 7 : Liste de demandes de locations

CaRental

[Owners board](#)
[Rental board](#)
[Models](#)
[Makes](#)
[Blacklist](#)
[Users' list](#)
[Add user](#)

Hello Amine!
[Logout](#)

### Créer votre compte

Username/Nom

Prenom

Numero de telephone

Ville

Adresse e-mail

Mot de passe

Confirmer mot de passe

IsAdmin

Créer un compte

© 2020 - CarRental

Language:
Anglais (États-Unis)

Save

**Figure 8 : Ajout d'un utilisateur par un administrateur**

CaRental

[Owners board](#)
[Rental board](#)
[Models](#)
[Makes](#)
[Blacklist](#)
[Users' list](#)
[Add user](#)

Hello Amine!
[Logout](#)

[Car Rental](#) >
[EspaceLocataire](#) >
[GetUsers2](#)

### All Users

First Name	Last Name	Phone Number	Email	
Said	Rochdi	0685520770	said@gmail.com	<div>!</div> <div>Add To Blacklist</div>
Rayan	Rayan	0632514786	rayan@gmail.com	<div>!</div> <div>Add To Blacklist</div>
Amine	Chakib	0632514786	amine@gmail.com	<div>!</div> <div>Add To Blacklist</div>
Moha	Moha	0632514786	moha@gmail.com	<div>!</div> <div>Add To Blacklist</div>
Saad	Benani	0658457415	said@gmail.com	<div>!</div> <div>Add To Blacklist</div>

© 2020 - CarRental

Language:
Anglais (États-Unis)

Save

**Figure 9 : Gestion des utilisateurs**

[Owners board](#)
[Rental board](#)
[Models](#)
[Makes](#)
[Blacklist](#)
[Users' list](#)
[Add user](#)

Hello Amine!
[Logout](#)

[Car Rental](#) >
[EspaceLocataire](#) >
[GetUsers](#)

## blacklist

First Name	Last Name	Phone Number	Email	
Chakib	Amine	0658457415	chakib@gmail.com	<a href="#">Remove from blacklist</a>

© 2020 - CarRental

Language:
Anglais (États-Unis)
[Save](#)

Figure 10 : Liste noire

[Owners board](#)
[Rental board](#)
[Models](#)
[Makes](#)
[Blacklist](#)
[Users' list](#)
[Add user](#)

Hello Amine!
[Logout](#)

[Car Rental](#) >
[Models](#) >
[Index](#)

## Models

[Search](#)

[+ Create New](#)

Model Name	Model Make	
Kangoo	Renault	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Details</a>
Q5	Audi	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Details</a>
RS7	Audi	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Details</a>
Q8	Audi	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Details</a>
GLC	Mercedes	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Details</a>
urus	Lamborghini	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Details</a>
Huracan	Lamborghini	<a href="#">Edit</a> <a href="#">Delete</a> <a href="#">Details</a>

© 2020 - CarRental

Language:
Anglais (États-Unis)
[Save](#)

Figure 11 : Gestion des modèles

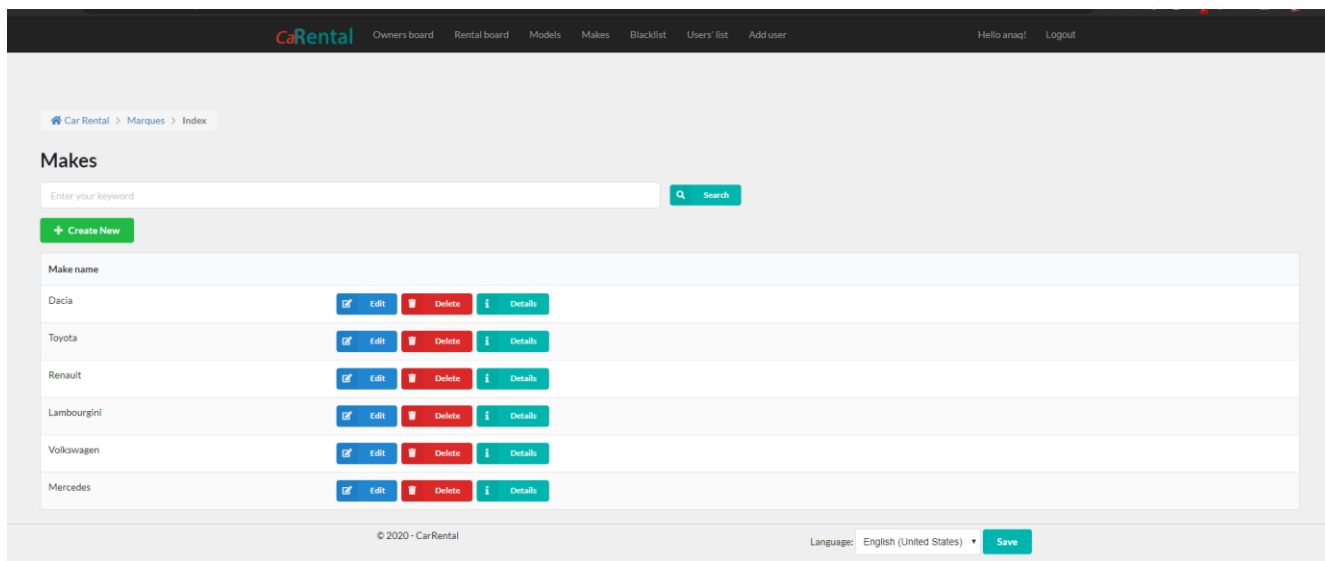


Figure 12 : Gestion des marques

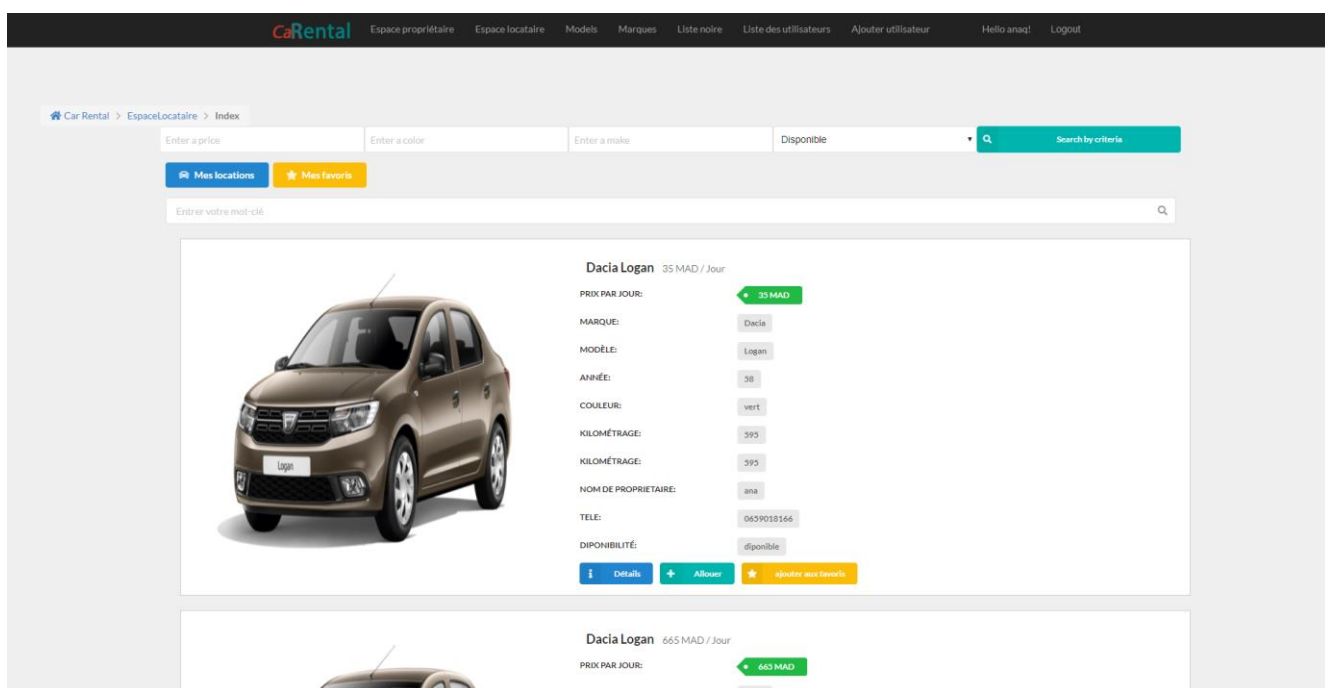


Figure 13: Espace Locataire (Francais)

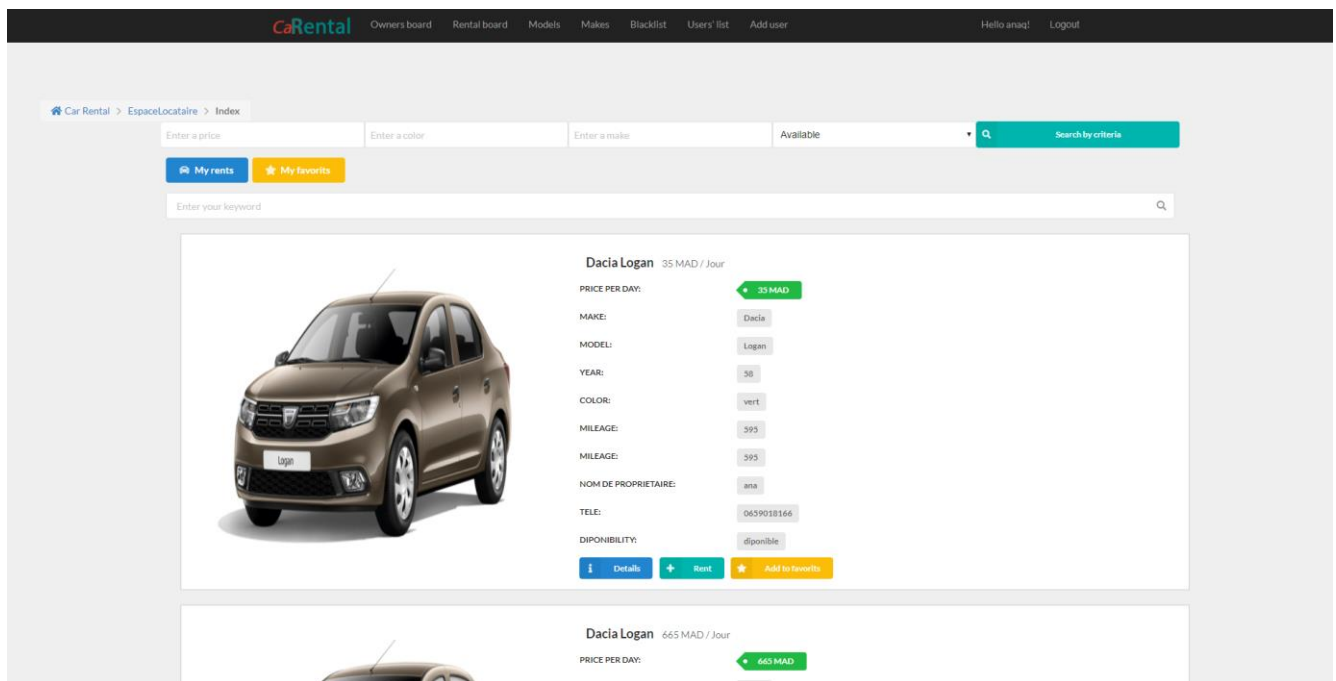


Figure 14: Espace Locataire (English)

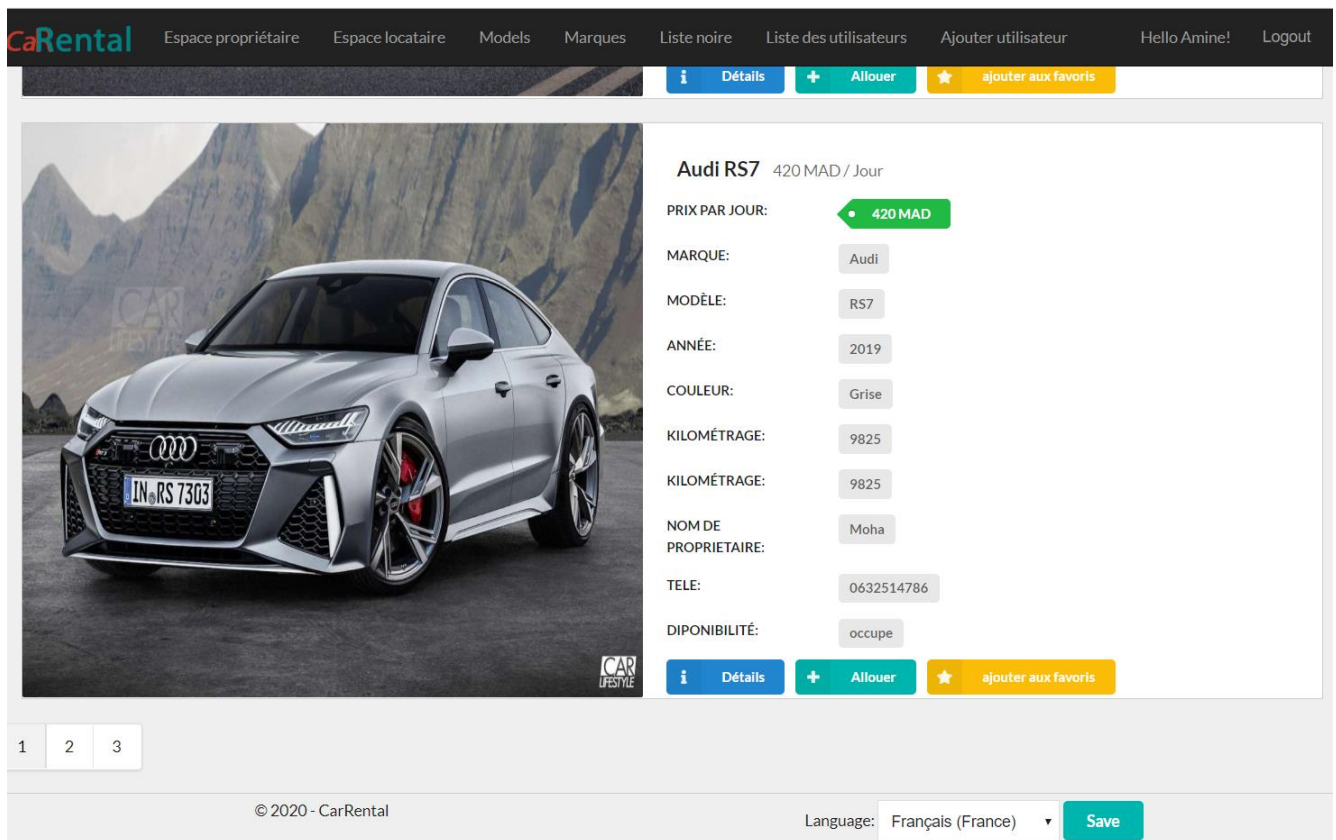



Figure 15 : Système de pagination

Car Rental

Owners boardRental boardModelsMakesBlacklistUsers listAdd user

Hello anaqiLogout

Car Rental > EspaceLocataire > favorites



Dacia Logan35 MAD / Jour

PRICE PER DAY:35 MAD

MAKE:Dacia

MODEL:Logan

YEAR:2018

COLOR:vert

MILEAGE:395

MILEAGE:395

NOM DE PROPRIETAIRE:ana


TELE:0659018166

DIPONIBILITE:disponible

Details

Ajouter

Rajouter favoris



Dacia Logan665 MAD / Jour

PRICE PER DAY:665 MAD

MAKE:Dacia

MODEL:Logan

YEAR:1982

COLOR:Rouge

Figure 16 : Gestion de favoris

22