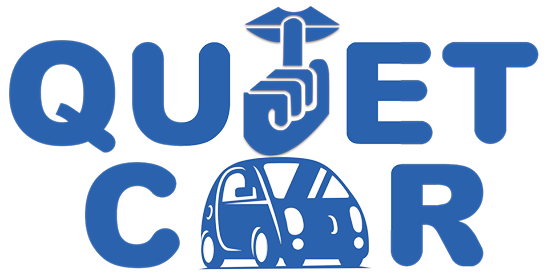


**Cahier de conception QuietCar**

 Le covoit’ sans blabla

**Rédigé par MANISCALCO Marie-Amélie, POUSSARD Sebastien, RIOUCH Badr**

Formation STRI Année 2019

**TABLE DES MATIÈRES**

[INTRODUCTION 3](#_Toc9712976)

[I. Architecture 4](#_Toc9712977)

[1. Architecture des fichiers 4](#_Toc9712978)

[2. Architecture HTML (gabarits) 5](#_Toc9712979)

[3. Contraintes 6](#_Toc9712980)

[a. Images 6](#_Toc9712981)

[b. Petit groupe & DevOps 6](#_Toc9712982)

[II. Constitution des pages 8](#_Toc9712983)

[1. Présentation des pages 8](#_Toc9712984)

[2. Charte graphique 13](#_Toc9712985)

[a. Le logo 13](#_Toc9712986)

[b. Typologie 14](#_Toc9712987)

[c. Couleurs 15](#_Toc9712988)

[d. Images 15](#_Toc9712989)

[III. UML 17](#_Toc9712990)

[1. Diagramme de cas d’utilisation 17](#_Toc9712991)

[a. Inscription 17](#_Toc9712992)

[b. Authentifier 18](#_Toc9712993)

[c. Rechercher un covoiturage 18](#_Toc9712994)

[d. Réserver un covoiturage 19](#_Toc9712995)

[e. Proposer un covoiturage 19](#_Toc9712996)

[f. Voir un profil 19](#_Toc9712997)

[g. Gérer un covoiturage 20](#_Toc9712998)

[h. Annuler une réservation 20](#_Toc9712999)

[i. Noter un covoiturage 20](#_Toc9713000)

[j. Supprimer un compte 21](#_Toc9713001)

[k. Supprimer un trajet 21](#_Toc9713002)

[2. Diagrammes de séquence 23](#_Toc9713003)

[a. Réservation d’un covoiturage 23](#_Toc9713004)

[b. Réservation d’un covoiturage 24](#_Toc9713005)

[3. Diagramme de navigation 25](#_Toc9713006)

[IV. Bases de données 30](#_Toc9713007)

[1. MCD 30](#_Toc9713008)

[2. Dictionnaire de données 30](#_Toc9713009)

[V. Choix technologiques 32](#_Toc9713010)

[CONCLUSION 34](#_Toc9713011)

[I. TABLE DES ILLUSTRATIONS 35](#_Toc9713012)

INTRODUCTION

Dans le but de réaliser un site de covoiturage, le cahier des charges réalisé précédemment présentait tout d’abord l’analyse des besoins, et les contraintes de programmation liées aux fonctionnalités obligatoires du site. Ensuite, une arborescence du site était présentée afin de définir la liaison entre les pages. Puis, une solution avait été proposée, chacune des pages avait été détaillée. La dernière partie du cahier des charge contenait un MCD et un MLD correspondant aux bases de données du site.

Ce cahier de conception présente dans un premier temps l’architecture du site c’est à dire l’arborescence des fichiers contenant le code de l’application puis le gabarit des pages HTML et les contraintes que cela engendre. Suite à la réalisation du cahier de conception, le développement du site web a pu commencer. Dans une première partie nous présenterons non pas les maquettes mais les pages réelles du site. Cette partie contient aussi la charte graphique du site. Une partie UML présente chaque fonctionnalité du site . Enfin les bases de données ont été modifiées pendant le développement elles ont donc été modélisées différement.

# **Architecture**

## Architecture des fichiers

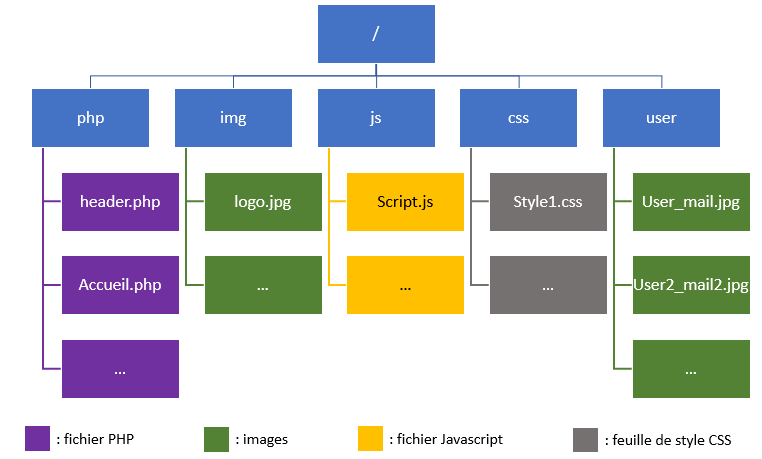


Figure : Architecture des fichiers

La figure suivante schématise l’architecture des fichiers :

La racine contient 5 dossiers :

* Php : contient les divers fichiers PHP.
* Img : contient les images de base du site web (logo, icones, …).
* Js : contient les script JavaScript.
* Css : contient les feuilles de styles CSS appliqués aux pages.
* User : contient les images des profils des utilisateurs, celle-ci sont nommées par rapport à l’identifiant de l’utilisateur c’est-à-dire son adresse mail où le « @ » a été remplacé par un « \_ ».

## Architecture HTML (gabarits)

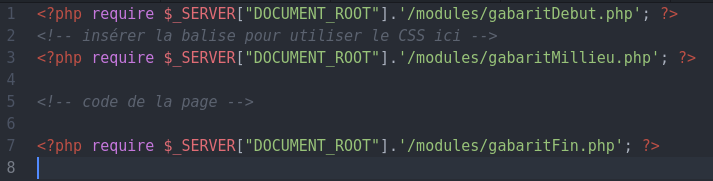


Figure : gabarit

Les pages PHP sont architecturés de la même manière et utilisent des modules :

* Gabarit début : Cette page contient la feuille de style principale qui s’applique à toutes les pages et le code vérifiant que l’utilisateur est connecté au site.
* Gabarit Milieu : cette page contient le menu haut du site ainsi que la partie rechercher un covoiturage sur la gauche
* Gabarit Fin : cette page contient la partie droite proposer ainsi que le footeur de la page.

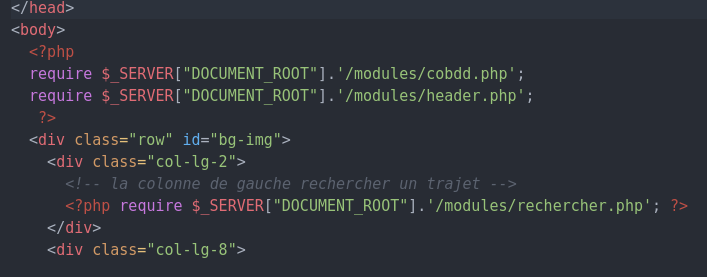


Figure : code du gabarit milieu

Dans la figure précédente nous avons l’exemple du code du gabarit milieu qui va lui-même importer 2 pages PHP :

* Cobdd.php qui sera responsable de la connexion à la base de données.
* Header.php qui sera le code du menu haut de l’application.

## Contraintes

### Images

Les images des profils utilisateurs ont une taille 180x180 pixels, avec différentes extensions acceptables : jpg,png,jpeg. Les images sont renommées par le système pour prendre le mail de l’utilisateur avec le quelle il s’est enregistré afin de retrouver l’image correspondant au profil.

Le logo du site est affiché avec une taille de 300x300 pixels.

### Petit groupe & DevOps

Le groupe de travail est composé de seulement 3 étudiants, ainsi pour rendre un travail de qualité nous avons choisis 2 stratégies :

* La première visait à limiter au départ le nombres de fonctionnalités que nous devions développer, nous nous sommes concentrés sur les fonctionnalités essentielles et nous en avons apporté de nouvelles par la suite.
* Une architecture DevOps (décrite dans la figure suivante) a permis d’optimiser notre temps de travail et ainsi pallier au manque de ressources humaines.

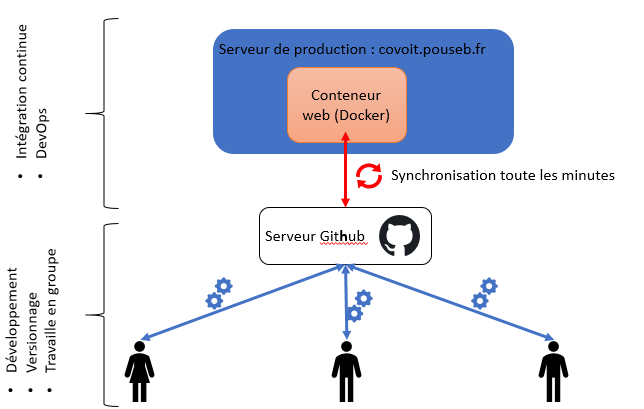


Figure : Architecture DevOps

Sur la figure suivante nous pouvons voir que nous nous appuyons sur plusieurs technologies.

La technologie Git nous permets ainsi de synchroniser notre travail et de le versionner, nous utilisons pour cela les serveurs de Github.

En parallèle nous avons mis en place un jeu de conteneur sous Docker pour accueillir un stack web (Conteneur Nginx, Conteneur PHP, Conteneur PostgreSQL et conteneur PgAdmin).

Les données de ce serveur sont synchronisées toutes les minutes avec le serveur Github.

Ainsi nous pouvons visualiser facilement le travail du groupe tout en travaillant avec les mêmes technologies et en utilisant la même base de données. Notre travail se fera donc en intégration continue grâce aux méthodes DevOps.

# **Constitution des pages**

## Présentation des pages

Cette partie présente les maquettes des pages du site. Chaque page en mode connecté présente une barre de menu avec un logo QuietCar un outil de recherche pour retrouver les utilisateurs, le nom de la page et un module présentant le nom de l’utilisateur connecté et sa photo.

On trouve aussi sur chaque page, à droite un formulaire de saisie pour rechercher un trajet où il faut entrer le lieu de départ et d’arrivée, la date l’heure et le nombre de bagages. À gauche le formulaire permet de proposer un trajet en entrant les mêmes informations que pour la recherche de trajet.

La première page du site est la page de connexion. Comme on le voit ci-dessous, il y a un premier module pour renseigner le login et un second pour le mot de passe. L’utilisateur a aussi la possibilité de sauvegarder ses identifiants sur la page pour ne pas avoir à les renseigner à chaque ouverture du site à l’aide d’une checkbox. Une fois que la personne a renseigné son identifiant et son mot de passe il clique sur connexion et si l’identifiant est le bon alors il accède aux fonctionnalités du site.

Si l’utilisateur n’a pas de compte et qu’il arrive sur cette page, il peut alors cliquer le bouton inscription et il accède alors à un formulaire qui lui permet de créer un compte QuietCar.



Figure : Page de connexion

Le formulaire d’inscription correspond à la page ci-dessous. Il faut tout d’abord entrer un nom, un prénom puis le genre de la personne. Ensuite, pour que les utilisateurs soient joignables, ils doivent renseigner un numéro de téléphone et une adresse mail qui leur servira également d’identifiant pour se connecter. Puis l’utilisateur doit entrer un mot de passe. Il y a ensuite des paramètres optionnels comme l’ajout d’une description et d’une photo. Une fois les informations du formulaire complétées, alors l’utilisateur clique sur le bouton «je m’inscris » s’il n’y a pas d’erreur le compte est créé et l’utilisateur est renvoyé sur la page de connexion où il entre ses nouveaux identifiants afin de pouvoir réserver ou proposer un trajet.



Figure : Page d'inscription

Lorsque l’utilisateur a un compte, il peut aller sur son profil dont la page est présentée ci-dessous. Ce dernier présente la photo et la description de la personne si elle les a entrés. D’autres informations sont affichées comme le nom le prénom de la personne, son numéro de téléphone et le type de sa voiture. Le profil indique aussi la moyenne des notes obtenues par le conducteur à coté de sa photo de profil et au-dessous du nombre de trajets qu’il a conduit ou qu’il a demandé.

En dessous de la présentation de la personne est affiché les trajets qu’il propose en indiquant la ville de départ et d’arrivée, la date, l’heure et le nombre de places encore disponibles.

Depuis son profil l’utilisateur peut aussi effectuer un changement de mot de passe.

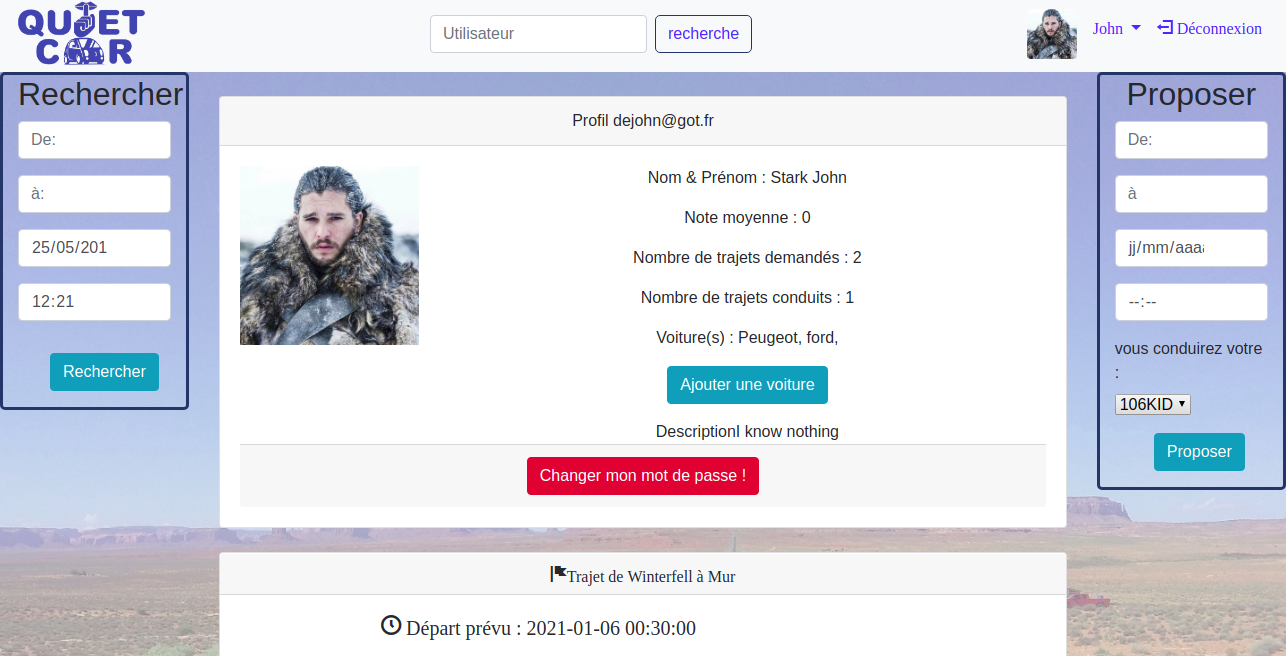


Figure : Page de profil

Lorsque l’utilisateur arrive sur le site, il se trouvera sur la page d’accueil, celle-ci contient les derniers trajets proposés par les utilisateurs.

Le haut de la page contient le logo de QuietCar, une barre de recherche d’user et sur la droite des informations concernant le profil connecté de l’utilisateur, il pourra consulter son profil ou se déconnecter depuis cet endroit, et sera sur chaque page du site.

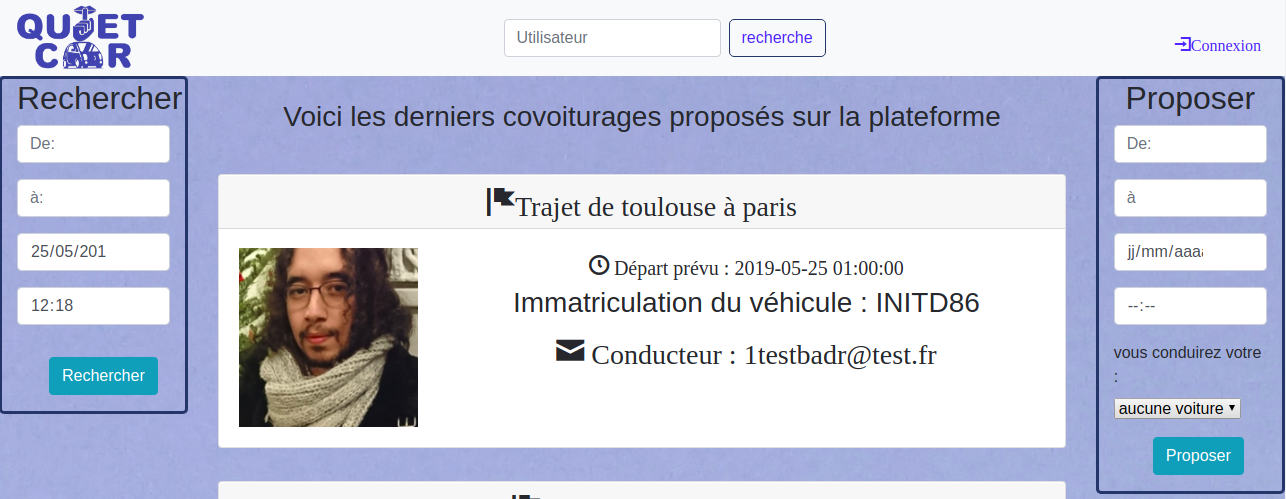


Figure  : Page d’accueil

Depuis son profil, l’utilisateur peut aussi accéder à un historique de ses covoiturages, classés par date, il peut les annuler depuis cette page et voir leur état (combien de passagers ont réservé, combien de bagages seront stockés, etc..).C’est aussi ici que le conducteur acceptera ou non les passagers.

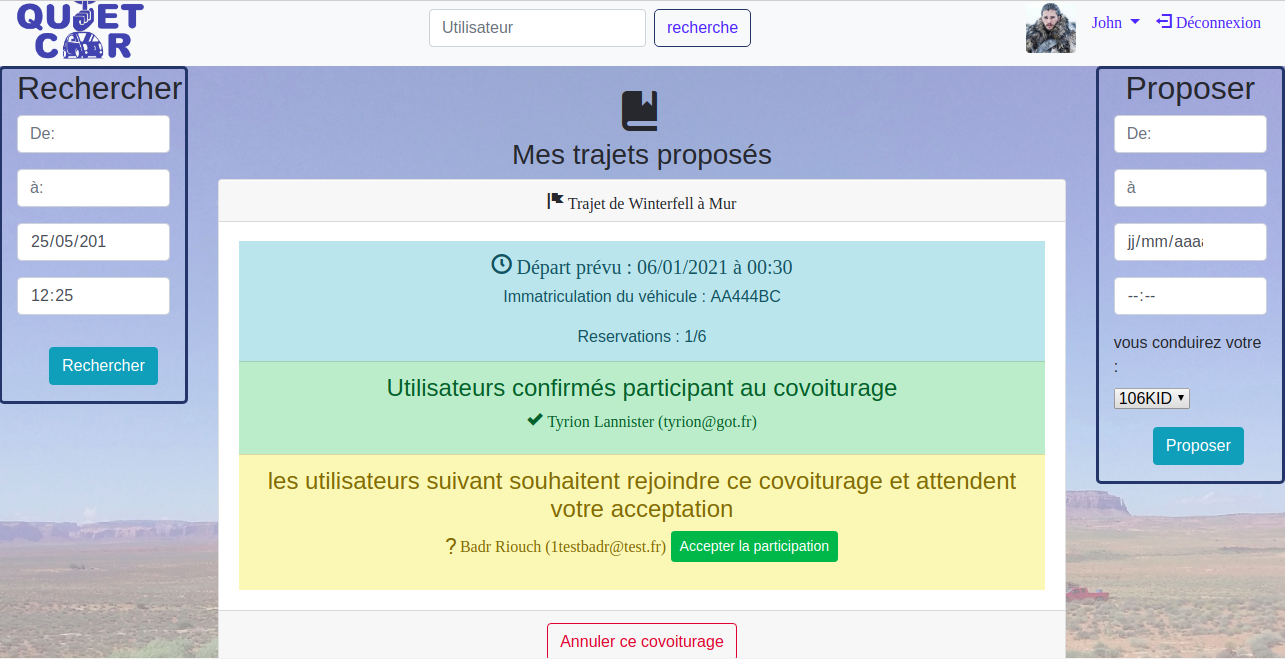


Figure : Mes Covoiturages

Lorsque l’utilisateur effectue une recherche selon le nom d’un utilisateur, les noms d’utilisateur les plus semblables à celui-ci apparaitront classés selon leur pertinence (exemple : si l’utilisateur cherche un utilisateur nommé « bernard » il apparaissent les utilisateurs « bernard tapie », « bernardo del gateau », « berduche la capuche ».

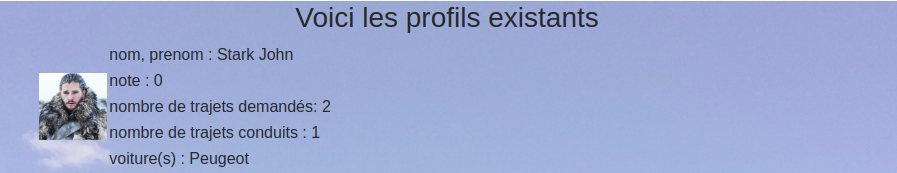


Figure  : Résultats de la recherche d'utilisateur

Idem pour la recherche de covoiturages.

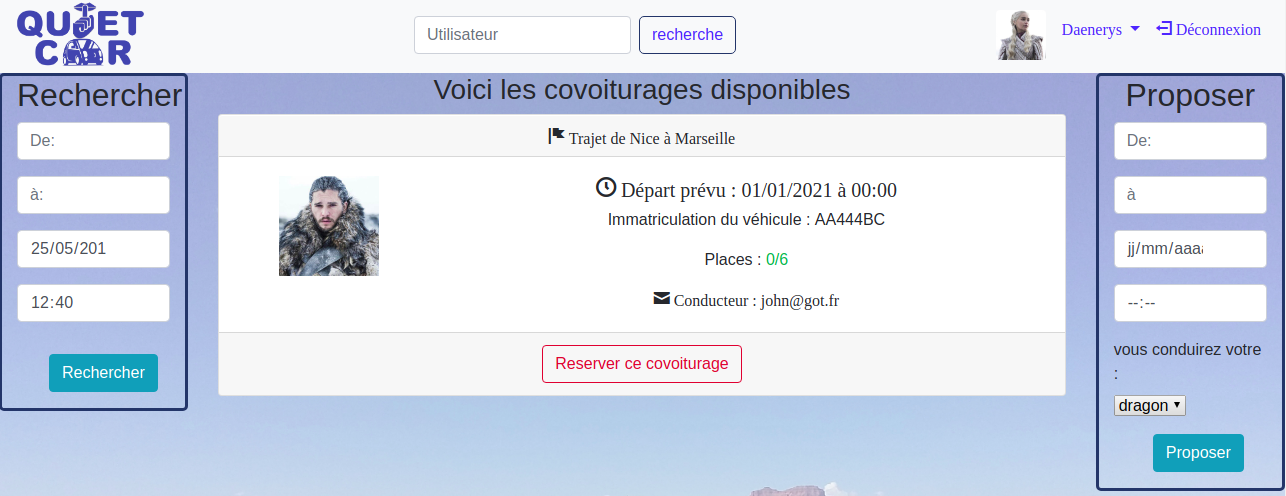


Figure  : Résultats de la recherche d'un covoiturage

Lorsque l’utilisateur aura créé un covoiturage via le menu sur la droite de la page, il sera renvoyé sur une page de création de covoiturage sur laquelle il aura le lieu de départ et d’arrivée, la date et l’heure de départ, les places disponibles et la limite de bagages qu’il aura entré, de plus il aura l’option de confirmer ou d’annuler son choix.

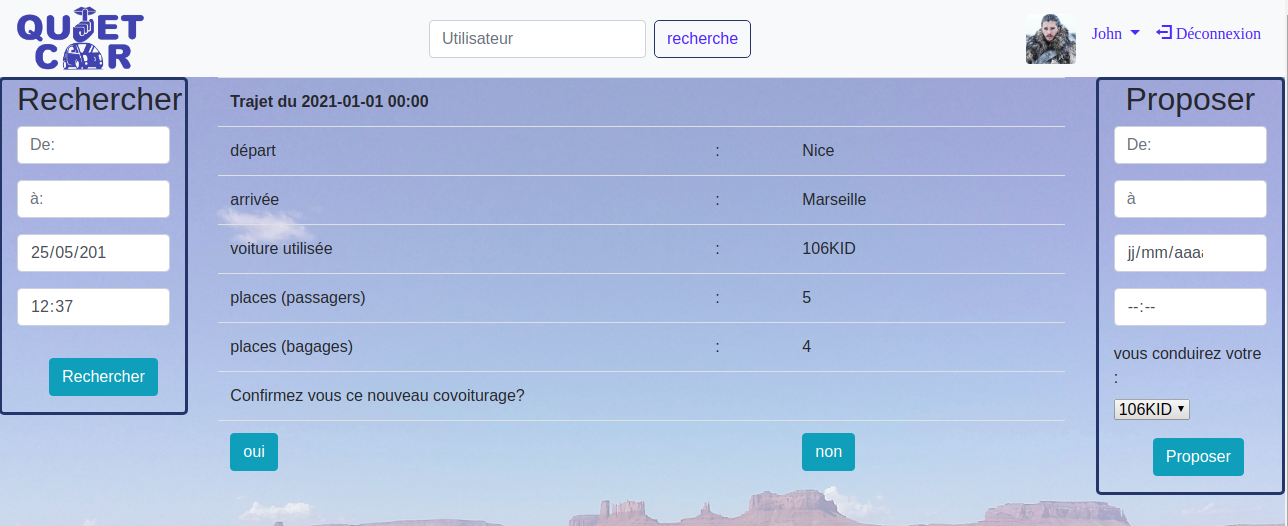


Figure : création de covoiturage

Lorsque l’utilisateur aura sélectionné un covoiturage, il sera renvoyé sur une page de confirmation de reservation sur laquelle il aura le lieu de départ et d’arrivée, la date et l’heure de départ, les places disponibles et la limite de bagages restants, de plus il aura l’option de confirmer ou d’annuler son choix.

## Charte graphique

### Le logo

Le logo est l’élément central de l’identité du site. C’est une image qui reste simple et facilement mémorisable. Afin de coordonner la communication du site il est nécessaire d’expliquer la signification du logo.

Celui-ci représente l’appellation du site de covoiturage : Quiet Car.

Le mot « quiet » est Illustré par un sigle de silence avec une bouche et une main dans le haut centre du logo, il remplace le i de quiet. Pour le terme « car » le A est remplacé par le sigle d’une voiture dans le bas centre du logo. Nous avons choisi un slogan en cohérence avec l’appellation que nous avons choisi : « le covoit’ sans blabla ».

Le logo et le slogan auront pour couleur du bleu céruléen (code #357AB7). Cette couleur reflète l’idée du voyage et des découvertes. C’est aussi le symbole de la vérité et de la loyauté en accord avec les principes des passagers et conducteurs du site de covoiturage. De plus, le bleu est une couleur universelle et facilement déclinable sur toutes les pages du site.

Le logo sera sur toutes les pages du site dans le coin à gauche de la barre de menu car ce sera un bouton de retour vers la page d’accueil.

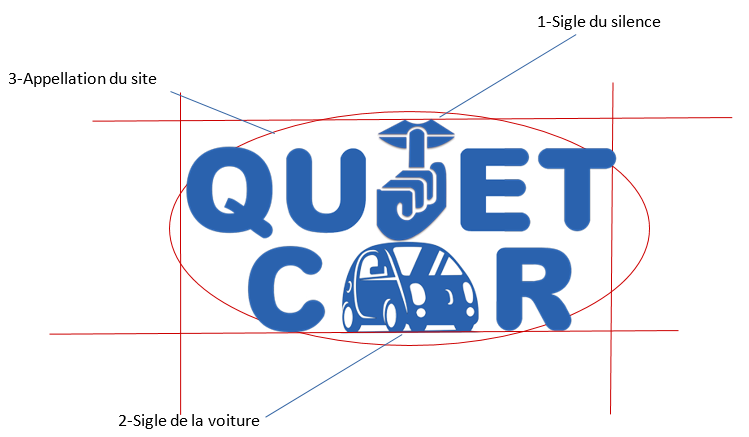


Figure : logo de l'application

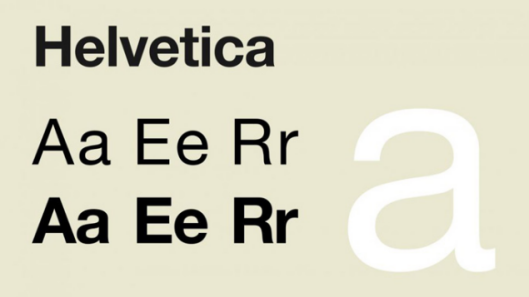
Des espaces de sécurité sont définis sur le schéma précédent. À l’intérieur de cette zone aucun image et aucun texte ne pourra être ajouté.

### Typologie

La typographie du site est en accord avec ses objectifs ainsi qu’avec son logo : voyager avec simplicité et dans le respect des engagements pris par le passager et le conducteur.

C’est donc la police Helvetica en taille 12px qui a été choisie pour sa simplicité et pour sa clarté. Elle a été réalisée par Max Miedinger en 1957, il souhaitait atteindre une certaine harmonie tout en respectant la neutralité. En effet cette police peut être appliquée pour de nombreux usages et est très utilisée dans le monde.

Le titre des pages a pour taille de police 20 px. Les sous titres sont de taille 15px en Italique.



Exemple de Titre des pages : Connexion

Exemple de sous-titre des pages : *Réservez un trajet*

Exemple de la corp du site : Bienvenue sur QuietCar

### Couleurs

Le thème couleur sélectionné et accord avec l’idée de voyage et d’ouverture vers de nouvelles horizons. Il respecte aussi le thème du logo.



Blanc

Bleu clair

Bleu foncé

Bleu medium

Figure : thème couleur choisi

Les pages sont composées d’un maximum de trois couleurs différentes, le gris, le noir et le blanc ne comptant pas comme des couleurs. La barre du menu ainsi que les différents boutons et les sous titres sont bleu medium (code : #226597). Les cadres des formulaires ou du profil sera en bleu foncé (code : #113F67). Le bouton de déconnexion sera bleu clair.

### Images

Le choix des images fait partie des outils amenant les conducteurs et les passagers à vouloir voyager.<http://frankryckewaert.com/la-route/>.

Dès lors, l’image de fond du site est la photo de la route qui mène à Monument Valley que l’on peut voir en arrière-plan. Ce paysage très vaste et très ouvert, invite les utilisateurs à voyager. Cette photo a été réalisée par les membres du groupe, elle est donc présente sur aucun autre site et n’est pas accessible sur google image.



Figure : image de fond du site

L’image de fond n’est pas la seule photo du site puisque chaque utilisateur aura sa photo de profil s’il le souhaite. Ces photos sont visibles depuis n’importe quelle image du site sur la barre de menu dans un petit cadre en haut à droite. Elles sont aussi visibles lorsqu’un utilisateur effectue une recherche de trajet. Ces images sont au format d’une photo d’identité.

# **UML**

## Diagramme de cas d’utilisation

La figure suivante représente le diagramme de cas d’utilisation qui seront chacun détaillés par la suite.

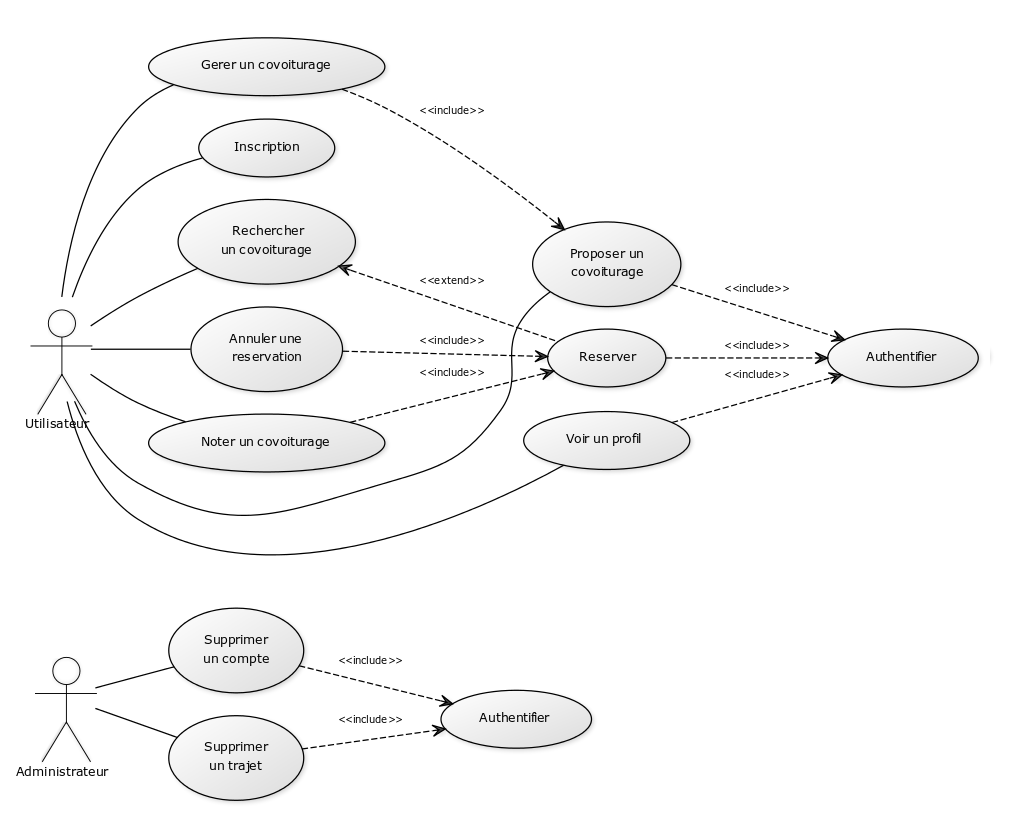


Figure 16 : Diagramme de cas d'utilisation

### Inscription

* Acteur principal : Utilisateur
* Acteur secondaire : aucun
* Objectif : Permettre à un utilisateur de s’inscrire sur le site afin qu’il puisse accéder aux fonctionnalités réservées aux inscrits (proposer et réserver des covoiturages).
* Préconditions : l’utilisateur n’est pas authentifié et n’existe pas déjà dans la base de données.
* Postconditions : Il est possible de s’authentifier avec l’utilisateur créer ; la base de données a été mise à jour.
* Scénario nominal :
  1. L’utilisateur souhaite s’inscrire afin de profiter de réserver un covoiturage.
  2. L’utilisateur va sur le page d’inscription et remplit le formulaire.
  3. Si le formulaire est correct alors l’utilisateur peut finaliser l’inscription, la base de données est mise à jour.

### Authentifier

* Acteur principal : Utilisateur et Administrateur
* Acteur secondaire : aucun
* Objectif : Permettre à un utilisateur ou à un administrateur de s’authentifier sur le site afin qu’il puisse accéder aux fonctionnalités réservées aux inscrits : proposer et réserver des covoiturages pour un utilisateur et supprimer un compte ou un trajet pour l’administrateur.
* Préconditions : l’utilisateur n’est pas authentifié et existe pas déjà dans la base de données (une inscription a été réalisée). Le compte administrateur ne nécessite pas d’inscription, il est défini à la base.
* Postconditions : aucune.
* Scénario nominal :
  1. L’utilisateur souhaite réserver un covoiturage.
  2. L’utilisateur s’authentifie.
  3. Si l’authentification réussie, l’utilisateur pourra accéder à la réservation.

### Rechercher un covoiturage

* Acteur principal : Utilisateur
* Acteur secondaire : aucun
* Objectif : Permettre à un utilisateur de visualiser les covoiturages qui sont proposés sur le site.
* Préconditions : aucune
* Postconditions : aucune.
* Scénario nominal :
  1. L’utilisateur cherche un covoiturage
  2. L’utilisateur se rend sur le page dédié et entre ses critères de recherches
  3. La liste des covoiturages apparaît.

### Réserver un covoiturage

* Acteur principal : Utilisateur
* Acteur secondaire : aucun
* Objectif : Permettre à un utilisateur de réserver un covoiturage.
* Préconditions : l’utilisateur doit être authentifié, le covoiturage sélectionné a encore des places disponibles
* Postconditions : le covoiturage est réservé, le covoiturage voit son nombre de places disponible réduit.
* Scénario nominal :
  1. L’utilisateur souhaite réserver un covoiturage.
  2. L’utilisateur s’authentifie.
  3. L’utilisateur recherche un covoiturage.
  4. L’utilisateur réserve un covoiturage.

### Proposer un covoiturage

* Acteur principal : Utilisateur
* Acteur secondaire : aucun
* Objectif : Permettre à un utilisateur de proposer un covoiturage.
* Préconditions : l’utilisateur doit être authentifié.
* Postconditions : le covoiturage est créé.
* Scénario nominal :
  1. L’utilisateur souhaite proposer un covoiturage.
  2. L’utilisateur s’authentifie.
  3. L’utilisateur proposer un covoiturage en renseignant les champs nécessaires (Ville de départ, d’arrivée …).
  4. Le covoiturage est disponible sur la plate-forme.

### Voir un profil

* Acteur principal : Utilisateur
* Acteur secondaire : aucun
* Objectif : Permettre à un utilisateur de voir le profil d’un utilisateur.
* Préconditions : l’utilisateur doit être authentifié.
* Postconditions : aucune.
* Scénario nominal :
  1. L’utilisateur souhaite avant de réserver se renseigner sur un utilisateur.
  2. L’utilisateur s’authentifie.
  3. L’utilisateur peut visualiser le profil d’un utilisateur.

### Gérer un covoiturage

* Acteur principal : Utilisateur.
* Acteur secondaire : un autre utilisateur (optionnel).
* Objectif : Permettre à un utilisateur d’accepter ou de refuser des utilisateurs qui souhaitent réserver sur son covoiturage ainsi que d’annuler le covoiturage.
* Préconditions : l’utilisateur doit être authentifié, l’utilisateur a proposé un covoiturage.
* Postconditions : l’état du covoiturage est modifié.
* Scénario nominal :
  1. Un utilisateur demande à réserver un covoiturage.
  2. L’utilisateur qui à crée le covoiturage est altéré.
  3. Celui-ci va sur la page pour gérer le covoiturage et accepte la demande de l’utilisateur.
  4. Le covoiturage propose désormais une place de moins de libre.

### Annuler une réservation

* Acteur principal : Utilisateur
* Acteur secondaire : aucun
* Objectif : Permettre à un utilisateur d’annuler sa réservation sur un trajet.
* Préconditions : l’utilisateur doit être authentifié, l’utilisateur a réservé un covoiturage.
* Postconditions : l’utilisateur ne participe plus au covoiturage, une place est libérée.
* Scénario nominal :
  1. Un utilisateur ne veut plus participer à un covoiturage.
  2. L’utilisateur sélectionne le covoiturage en question et annule sa participation.
  3. L’utilisateur ne participe plus au covoiturage, une place est libérée.

### Noter un covoiturage

* Acteur principal : Utilisateur
* Acteur secondaire : Utilisateur conducteur
* Objectif : Permettre à un utilisateur qui a participé à un covoiturage de noter le trajet.
* Préconditions : l’utilisateur doit être authentifié, l’utilisateur a réservé un covoiturage, l’utilisateur a participé à un covoiturage.
* Postconditions : l’utilisateur conducteur a une note qui lui est attribuée qui vient moduler sa moyenne.
* Scénario nominal :
  1. Un utilisateur veut participer à un covoiturage.
  2. L’utilisateur s’inscrit à un covoiturage.
  3. L’utilisateur participe à un covoiturage.
  4. Une fois le voyage terminé, l’utilisateur attribue une note à l’utilisateur conducteur.

### Supprimer un compte

* Acteur principal : Administrateur
* Acteur secondaire : Utilisateur
* Objectif : Permettre à l’administrateur de supprimer un compte utilisateur.
* Préconditions : l’administrateur doit être authentifié, le compte utilisateur doit exister.
* Postconditions : le compte utilisateur est supprimé, celui-ci ne peut plus s’authentifier avec ce compte.
* Scénario nominal :
  1. L’administrateur souhaite supprimer un compte utilisateur.
  2. L’administrateur s’authentifie.
  3. L’administrateur supprime le compte utilisateur.

### Supprimer un trajet

* Acteur principal : Administrateur
* Acteur secondaire : Utilisateur
* Objectif : Permettre à l’administrateur de supprimer un compte utilisateur.
* Préconditions : l’administrateur doit être authentifié, le compte utilisateur doit exister.
* Postconditions : le compte utilisateur est supprimé, celui-ci ne peut plus s’authentifier avec ce compte.
* Scénario nominal :
  1. L’administrateur souhaite supprimer un compte utilisateur.
  2. L’administrateur s’authentifie.
  3. L’administrateur supprime le compte utilisateur.

## Diagrammes de séquence

### Réservation d’un covoiturage

La figure suivante décrit l’enchaînement d’opérations qui se déroulent quand un utilisateur souhaite réserver un covoiturage.

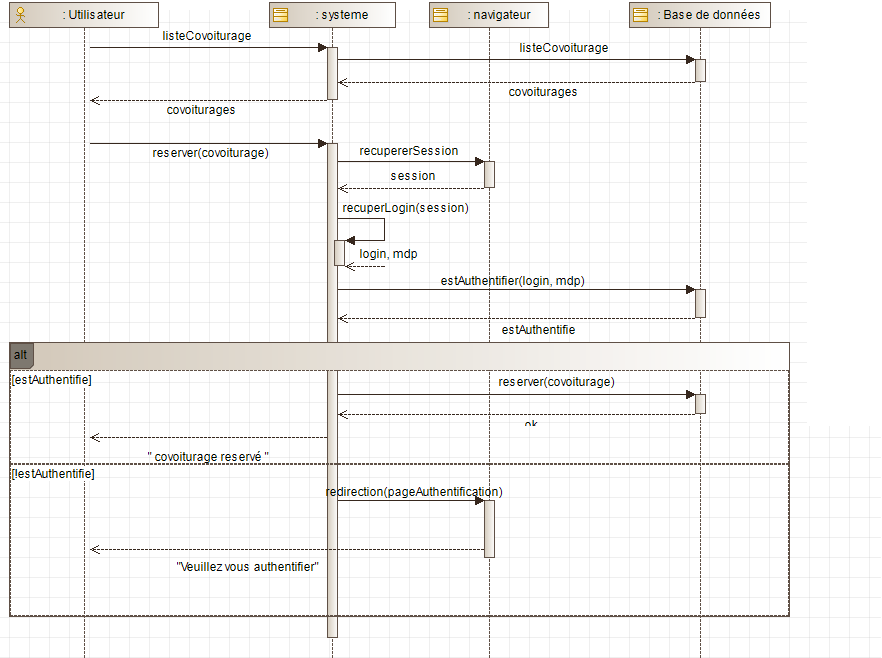


Figure : Diagramme de séquence (Réserver un covoiturage)

L’utilisateur va commencer par obtenir la liste des covoiturages qui sont disponibles, l’application va demander à la base de données cette liste et l’afficher à l’utilisateur.

L’utilisateur va ensuite sélectionner un covoiturage qu’il va vouloir réserver. L’application va récupérer la session et vérifier avec la base de données si l’utilisateur est correctement authentifié pour la suite.

Si l’utilisateur est correctement authentifié, l’application va modifier la base de données et informer l’utilisateur que la réservation c’est déroulé avec succès.

Si l’utilisateur n’est pas correctement authentifié, l’application va rediriger l’utilisateur vers la page d’authentification en lui demandant de s’authentifier.

### Réservation d’un covoiturage

La figure suivante décrit l’enchaînement d’opérations qui se déroulent quand un administrateur souhaite supprimer un utilisateur.

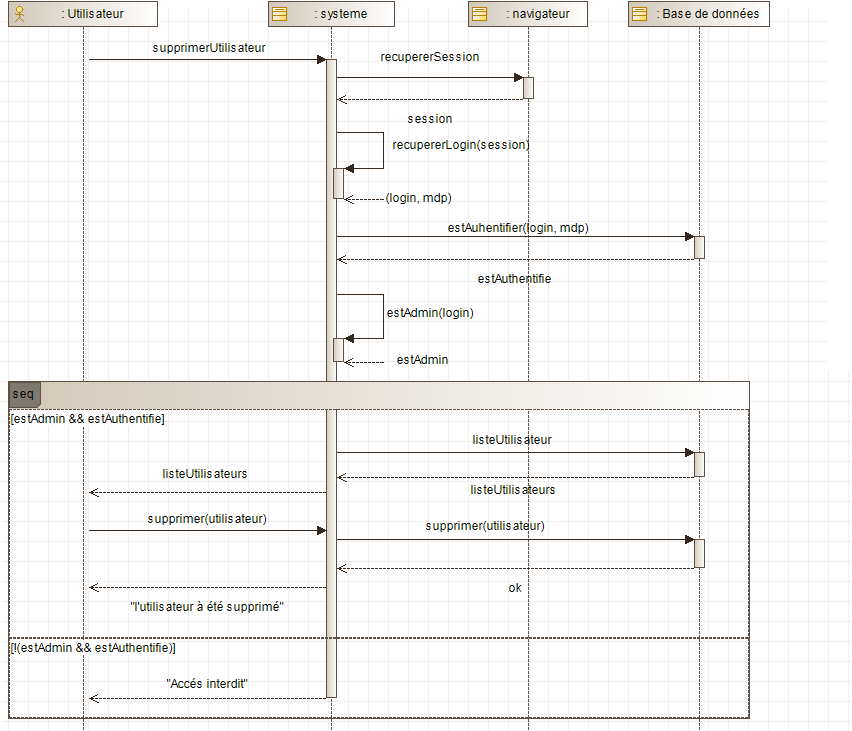


Figure : Diagramme de séquence de la suppression d'un utilisateur

Quand un Administrateur souhaite supprimer un utilisateur, l’application va récupérer la session du navigateur et vérifier avec la base de données si l’utilisateur est authentifié et si celui-ci est un administrateur.

Si l’utilisateur est authentifié et qu’il est un administrateur, l’application va récupérer la liste des utilisateurs de la base de données et la renvoyer à l’administrateur. L’administrateur sélectionne un utilisateur à supprimer, l’application va supprimer l’utilisateur de la base de données et informer l’administrateur du succès de l’opération.

Si l’utilisateur n’est pas authentifié ou qu’il n’est pas un administrateur alors l’application envoie un message indiquant que l’accès est interdit.

## Diagramme de navigation

La figure suivante décrit la navigation sur l’ensemble du site.

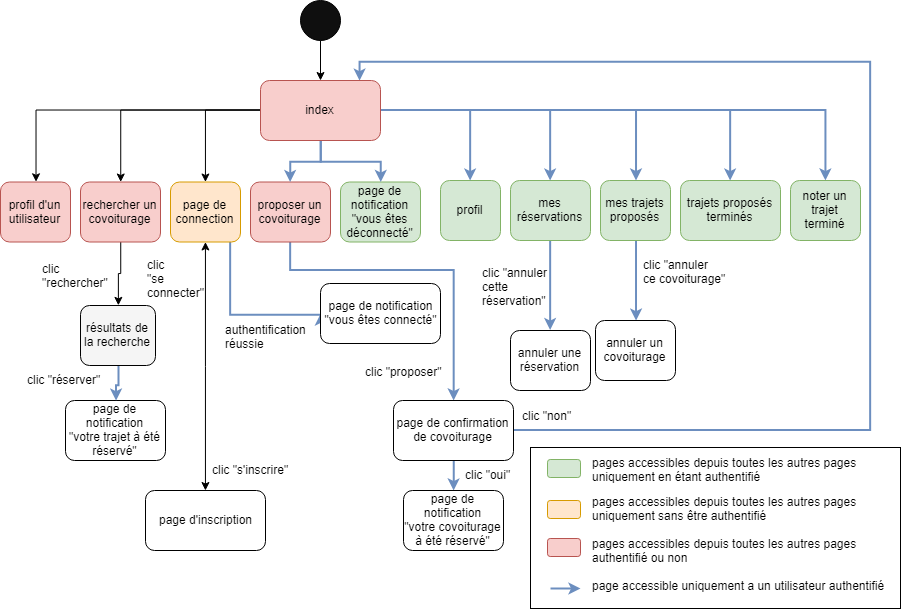


Figure : Diagramme de navigation

# **Bases de données**

## MCD

La figure suivante représente le Modèle Conceptuel de Données

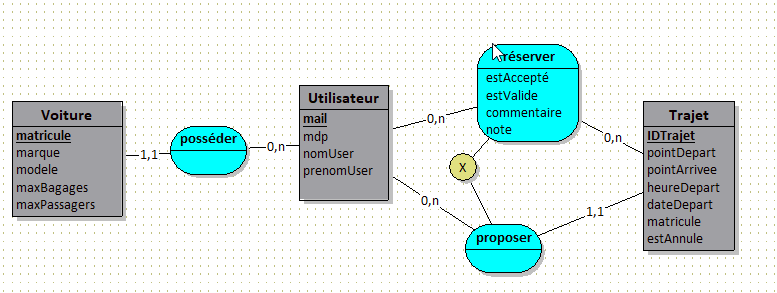


Figure : MCD

* **Utilisateur** = (mail, mdp, nomUser, prenomUser);
* **Trajet** = (IDTrajet, pointDepart, pointArrivee, heureDepart, dateDepart, estAnnule, #conducteur);
* **Voiture** = (matricule, marque, modele, maxBagages, maxPassagers, #proprietaireVoiture);
* **Réserver** = (#mail, #IDTrajet, estAccepte, estValideRes, Commentaire, note);

## Dictionnaire de données

Voiture

* Matricule (VARCHAR) : le matricule de la voiture
* Marque (VARCHAR) : la marque de la voiture
* Modele (VARCHAR) : le modèle de la voiture
* maxBagages (INT) : le reste des places de bagages dans la voiture
* maxPassagers (INT) : le reste des places pour passagers dans la voiture

Utilisateur

Mail (VARCHAR) : adresse mail de l’utilisateur

* Mdp (VARCHAR) : mot de passe de l’utilisateur
* NomUser (VARCHAR) : nom de l’utilisateur
* PrenomUser (VARCHAR) : prenom de l’utilisateur
* Ban (BOOLEAN) : définit le bannissement de l’utilisateur du site

Trajet

* IDTrajet (INT) : identifiant du trajet
* PointDepart (VARCHAR) : point de départ du trajet
* PointArrivee (VARCHAR) : point d’arrivée du trajet
* DateHeureDepart (DATE) : date et heure du départ
* EstAnnule (BOOLEAN) : définit l’état du trajet
* Matricule (VARCHAR) : matricule de la voiture utilisé pour le trajet

Réserver

* EstAccepte (BOOLEAN) : définit si la réservation a été acceptée par le conducteur
* EstValideRes (BOOLEAN) : définit si la réservation a été validée ensuite par le passager
* Commentaire (VARCHAR) : commentaire laissé par le passager
* Note(INT) : note laissé par le passager

# Choix technologiques

Tout d’abord, de nombreuses fonctionnalités du site demandent l’enregistrement de données. En effet, la partie connexion nécessite le stockage des identifiants et mot de passe de chaque utilisateur, la partie inscription et profil également car elle présentera des informations sur les personnes afin de pouvoir les contacter. Il faut ensuite pouvoir enregistrer et afficher tous les trajets en fonction de différents paramètres. Enfin, la note attribuée au conducteur doit aussi être enregistrée. Cet aspect du projet va donc nous contraindre à utiliser les bases de données pour pouvoir enregistrer toutes les informations sur les utilisateurs et sur les trajets. Pour cela nous utiliserons PostgreSQL.

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD. Il est conforme aux standards SQL et est reconnue pour être très stable.



Figure : PostgreSQL

Le client souhaite que nous développions une application web, pour ce faire nous utiliserons donc HTML5 qui est le standard pour les sites web.

Figure : HTML5

D’autre part, il nous faut un site web dynamique, avec diverses fonctions s’adaptant aux requêtes des clients. Afin de réaliser de telles fonctions il est nécessaire d’utiliser le langage PHP car il permet de communiquer avec les bases de données pour la connexion, l’inscription, l’affichage du profil ainsi que pour la recherche de trajets. Pour travailler l’architecture des pages nous devront utiliser le langage HTML dans lequel nous devrons intégrer l’appel aux fonctions PHP.



Figure :PHP

Le site nécessite un certain traitement ainsi que des interactions entre l’application web et l’utilisateur. Pour que les échanges soient plus confortables nous utiliserons le langage Java Script comme dans tous site web moderne, celui-ci permettra d’apporter une interactivité côté client et non côté serveur (animations, interactions …).



Figure : Javascript

Pour que QuietCar s’adapte correctement aux différents navigateurs du marché nous allons respecter les contraintes du W3C qui imposent le respect des standards de codage HTML



Figure : W3C

Enfin l’application Web doit être responsive design, c’est-à-dire qu’elle doit s’adapte aux différents supports pour le visualiser (Smartphone, Tablette, Ordinateur…) sans déformer le contenue. Pour résoudre ce problème nous utiliserons Bootstrap.



Figure : Bootstrap

# **CONCLUSION**

Ce document présente différents aspects du site. Tout d’abord, son architecture c’est à dire l’organisation des fichiers de développement, le gabarit HMTL des pages et les contraintes. Puis les aspects visuels sont abordés, les pages sont expliquées à l’aide d’une photo de chacune d’entre elles, la charte visuelle du site a été définie et doit être respectée également dans les différents documents de présentation du site. Nous présentons également les fonctionnalités du site définies sous UML.

Enfin nous avons abordé les choix technologiques ainsi que la modélisation des données sous forme de MCD, leur traduction et un dictionnaire de données.

L’architecture permettant d’accueillir l’application web a été mise en place et pourra être suivie de manière continue par l’enseignant à travers le site de production « https://covoit.pouseb.fr » et le github <https://github.com/sebastienPoussard/CovoitWeb>.

Ainsi, le site web QuietCar présente de nombreuses fonctionnalités comme la recherche de trajets, la proposition de trajets et la selection. L’utilisateur dispose d’un compte personnel avec un profil personnalisé qui permet de mettre en confiance les autres utilisateurs. Bien que de nombreuses fonctionnalités aient été développées certains points sont encore à aborder comme la sécurité du site ou encore la confirmation de trajets et de mots de passe par un envoi de mail.

# **TABLE DES ILLUSTRATIONS**

[Figure 1 : Architecture des fichiers 5](#_Toc9713025)

[Figure 2: gabarit 6](#_Toc9713026)

[Figure 3 : code du gabarit milieu 6](#_Toc9713027)

[Figure 4 : Architecture DevOps 8](#_Toc9713028)

[Figure 5: Page de connexion 9](#_Toc9713029)

[Figure 6: Page d'inscription 10](#_Toc9713030)

[Figure 7 : Page de profil 11](#_Toc9713031)

[Figure 8 : Page d’accueil 11](#_Toc9713032)

[Figure 9 : Mes Covoiturages 12](#_Toc9713033)

[Figure 10 : Résultats de la recherche d'utilisateur 12](#_Toc9713034)

[Figure 11 : Résultats de la recherche d'un covoiturage 13](#_Toc9713035)

[Figure 12 : création de covoiturage 13](#_Toc9713036)

[Figure 13 : logo de l'application 15](file:///\\VBOXSVR\share\Cahier-De-Conception-Mise-en-page.docx#_Toc9713037)

[Figure 14 : thème couleur choisi 16](#_Toc9713038)

[Figure 15 : image de fond du site 17](#_Toc9713039)

[Figure 16 : Diagramme de cas d'utilisation 18](#_Toc9713040)

[Figure 17 : Diagramme de séquence (Réserver un covoiturage) 24](#_Toc9713041)

[Figure 18 : Diagramme de séquence de la suppression d'un utilisateur 25](#_Toc9713042)

[Figure 19 : Diagramme de navigation 26](#_Toc9713043)

[Figure 20 : MCD 31](#_Toc9713044)

[Figure 21 : PostgreSQL 33](#_Toc9713045)

[Figure 22 : HTML5 33](file:///\\VBOXSVR\share\Cahier-De-Conception-Mise-en-page.docx#_Toc9713046)

[Figure 23 :PHP 34](#_Toc9713047)

[Figure 24 : Javascript 34](#_Toc9713048)

[Figure 25 : W3C 34](#_Toc9713049)

[Figure 26 : Bootstrap 34](#_Toc9713050)

# **Répartition et Vidéo de présentation**

Compte tenu des différentes taches à mettre en œuvre pour le projet (rédaction des différents documents, développement, mise en page, mise en place des moyens d’hébergement etc…) la répartition des taches au sein du groupe est la suivante :

Sébastien POUSSARD : 35%

Badr RIOUCH : 32,5%

Marie-Amélie MANISCALCO : 32,5%

Une vidéo de présentation peut être trouvée au lien suivant : <https://youtu.be/-TVvxsprbj4>