

CAHIER DES CHARGES QUIETCAR



« Le covoit' sans blabla »

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

QuietCar, site de covoiturage

Cahier des charges

État	En cours de rédaction
Auteurs	Maniscalco Marie-Amélie, Sébastien Poussard, Badr Riouch
Date	26/02/2018

L'objectif de ce document est de définir :

- Les besoins et contraintes
- Les différentes fonctionnalités de l'application
- Les croquis et l'arborescence du site

TABLE DES MATIÈRES

PRESENTATION DU PROJET	4
PROTAGONISTES	5
1. Equipe	5
2. Client	5
ANALYSE DE L'EXISTANT	6
I. Besoin et contraintes	7
1. Analyse des besoins	7
2. Contraintes	8
3. Architecture	9
II. Solutions proposés	11
1. Connexion et inscription	11
2. Les trajets	12
3. Système de paiements	14
III. Base de données	16
1. MCD	16
2. MLD	17
CONCLUSION	18
TABLE DES ILLUSTRATIONS	19

PRÉSENTATION DU PROJET

Dans le cadre de l'apprentissage du Web Dynamique et des bases de données, nous allons développer un site web de covoiturage multi-utilisateurs et multi-trajets. Celui-ci reprendra les principales fonctionnalités de Blablacar premier site de covoiturage en France.

L'application aura pour monnaie d'échange entre conducteurs et passagers un système de note lorsque le trajet est effectué. Chaque utilisateur disposera d'un compte et d'un profil visible par les autres. Il y aura ensuite un système de recherche des trajets disponibles en fonction de la ville de départ et de la ville d'arrivée.

Le site devra être fonctionnel pour le 23 Mai 2019 date à laquelle nous présenterons ses fonctions.

Le présent document décrit tout d'abord l'équipe et le client puis les besoins et les contraintes liées au projet en présentant l'arborescence du site. Enfin, les solutions trouvées et le résultat attendu seront présentés en illustrant chaque fonctionnalité de l'application.

PROTAGONISTES

1. Equipe

L'équipe est composée de 3 étudiants en 1^{er} année de STRI à savoir :

- Maniscalco Marie-Amélie : Membre du groupe le plus expérimentée dans la conception de site web, elle apportera au groupe son expérience technique dans les langages qui seront utilisés pour coder et permettra aux groupes de progresser ensemble.
- Riouch Badr : amateur de conception graphique et d'interface homme-machine, il apporte au groupe l'identité visuelle de notre site, ainsi qu'une certaine maîtrise de Javascript.
- Poussard Sébastien : par son expérience dans la gestion de projets professionnels il apportera le cadre organisationnel pour optimiser le travail du groupe ainsi qu'une maîtrise des langages de base de données.

Les différents membres du groupe participeront ensemble à tous les aspects du projet de sa conception à son codage tout en faisant monter en compétence l'équipe sur leur sujet de prédilection à travers des rendez-vous réguliers pour travailler ensemble.

2. Client

Les clients de ce projet seront les enseignants Mr BRET Martial et Mr MOJAHID Mustapha enseignants en STRI

.

ANALYSE DE L'EXISTANT

Il existe plusieurs sites de covoiturage en France. Laroueverte.com, Carpooling.fr ou encore Blablacar, ils regroupent tous les mêmes fonctions principales c'est à dire proposer ou rechercher un covoiturage. Les sites n'ont pas tous le même intérêt car certains proposent uniquement de courts trajets pour se rendre au travail et d'autres proposent de plus longs trajets en France.

Le plus connu des sites de covoiturage en France est Blablacar. Pour proposer un trajet ou bien rechercher un trajet l'utilisateur doit s'inscrire et remplir une liste des préférences pour trouver un covoiturage qui correspond au trajet qu'il souhaite effectuer mais aussi en fonction de ses centres d'intérêts et de ses exigences.

Le conducteur peut donc proposer un trajet en renseignant son itinéraire, son heure de départ et ces options comme non-fumeur, présence d'animaux ou encore le nombre de bagages. La personne qui cherche un trajet doit renseigner une ville de départ une ville d'arrivée et une date pour obtenir la liste des covoiturages disponibles ce jour. Elle peut ensuite consulter les exigences du conducteur et même visionner son profil pour en savoir plus sur sa personnalité mais aussi pour voir les commentaires et les notes laissées par les anciens passagers.

BlaBlaCar propose un système de paiement en ligne sécurisé.



FIGURE 1 : LOGO BLA BLA CAR

I. BESOIN ET CONTRAINTES

1. Analyse des besoins

Le site de covoiturage présente certaines fonctions obligatoires qui sont présentées dans cette partie. De nouvelles fonctionnalités pourraient être ajoutées en plus de celles-ci.

Tout d'abord, le site doit être accessible seulement par les utilisateurs disposant d'un compte. Donc, pour accéder aux fonctionnalités du site, le client devra s'authentifier, cela permettra de sécuriser les données des conducteurs (celui qui propose un trajet) et des demandeurs (ceux qui cherchent un trajet).

Si l'utilisateur n'a pas de compte il devra donc pouvoir en créer un. Pour cela il renseignera les principales informations nécessaires à la création d'un profil. Tous les utilisateurs auront un profil consultable par les membres du site. Cela permettra aux différentes personnes de se contacter et de consulter les trajets proposés. Cette partie de l'application va nécessiter la création d'une fonction de connexion, d'inscription ainsi que d'un système de publication pour générer le profil de l'utilisateur. Chaque personne aura la possibilité de modifier les informations publiées sur son profil.

Le conducteur qui propose un trajet doit pouvoir le renseigner en indiquant la ville de départ et la ville d'arrivée. Il doit aussi indiquer l'horaire auquel il souhaite partir ainsi que le nombre de places disponibles. Ces informations vont permettre au demandeur de covoiturage de voir si cela correspond à ce qu'il souhaite. Pour cela il doit pouvoir effectuer une recherche en fonction de la ville de départ et de la ville d'arrivée. Il devrait ensuite pouvoir consulter la liste des covoiturations qui partent de cette ville et qui vont jusqu'à la ville d'arrivée. La liste présenterait aussi les horaires de départ et le nombre de places disponibles. Le demandeur doit ensuite pouvoir sélectionner le covoiturage de son choix si celui-ci est encore disponible c'est à dire, s'il y a encore des places dans le véhicule. Mais, une fois le trajet choisi, le demandeur doit pouvoir annuler sa demande. Il sera donc nécessaire de créer une fonction d'ajout de trajet pour le conducteur, une fonction de recherche pour que le demandeur puisse choisir un trajet à sa convenance puis une fonction permettant au demandeur d'annuler son choix afin d'en alerter le conducteur.

Une fois le trajet effectué le demandeur doit payer le conducteur en lui attribuant une note. La note devrait être visible sur le profil du conducteur afin d'inciter ou non d'autres utilisateurs à effectuer des covoiturations avec la personne. Pour cette partie de l'application il faudrait une fonction de notation pour que le demandeur puisse attribuer

une note à son conducteur mais aussi une fonction effectuant la moyenne de toutes les notes reçues par un conducteur pour que celle-ci soit visible sur son profil.

2. Contraintes

Les principales fonctions que nous allons créer pour répondre aux besoins du client présentent certaines contraintes auxquelles nous allons devoir prêter attention.

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.



FIGURE 2 : LOGO POSTGRESQL

D'autre part, il nous faut un site web dynamique, avec divers 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 devons intégrer l'appel aux fonctions PHP.



FIGURE 3 : LOGO 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.

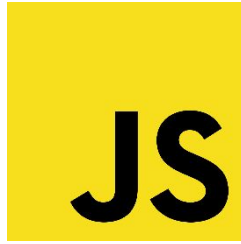


FIGURE 4 : LOGO JAVASCRIPT

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 5 : LOGO W3C

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



FIGURE 6 : LOGO BOOTSTRAP

3. Architecture

Après vous avoir présenté les fonctionnalités principales du site à réaliser, nous avons vu que nous allons devoir faire face à certaines contraintes. Nous allons maintenant étudier l'architecture du site que nous allons employer pour relier les fonctions et les pages.

La première page sur laquelle l'utilisateur arrive lorsqu'il accède au site sera la page de connexion. Sur celle-ci il pourra se connecter puis accéder au menu principal si ses identifiants sont corrects ou bien s'inscrire et il accèdera à la page d'inscription. Celle-ci renverra à la page de connexion une fois le nouvel utilisateur enregistré.

L'utilisateur connecté accède donc au menu principal où il peut rechercher un trajet ce qui l'enverra sur une page de recherche mais il peut aussi proposer un trajet et visualiser son profil ou bien se déconnecter.

Depuis son profil, l'utilisateur peut joindre la page de recherche de trajet ainsi que la page d'ajout de trajets. Il peut également se déconnecter.

L'utilisateur accède à la page de la recherche qui a été affichée après avoir renseigné une ville d'arrivée et de départ dans le module de recherche du menu principal. Depuis sa recherche, l'utilisateur doit pouvoir visualiser le profil du conducteur qui propose un covoiturage pour vérifier ses informations. Il peut ensuite revenir à sa recherche et valider sa demande de covoiturage. Depuis sa recherche le demandeur doit aussi pouvoir retourner au menu principal et se déconnecter.

Ensuite, lorsqu'un conducteur souhaite proposer un covoiturage il clique sur le bouton du menu principal qui propose l'ajout d'un trajet et il arrive sur une nouvelle page. Il peut alors renseigner son trajet et le publier ou bien retourner au menu principal et se déconnecter.

La fonction de déconnexion ainsi que le menu principal seront accessibles depuis toutes les pages du site sauf depuis le menu de connexion qui est la première page.

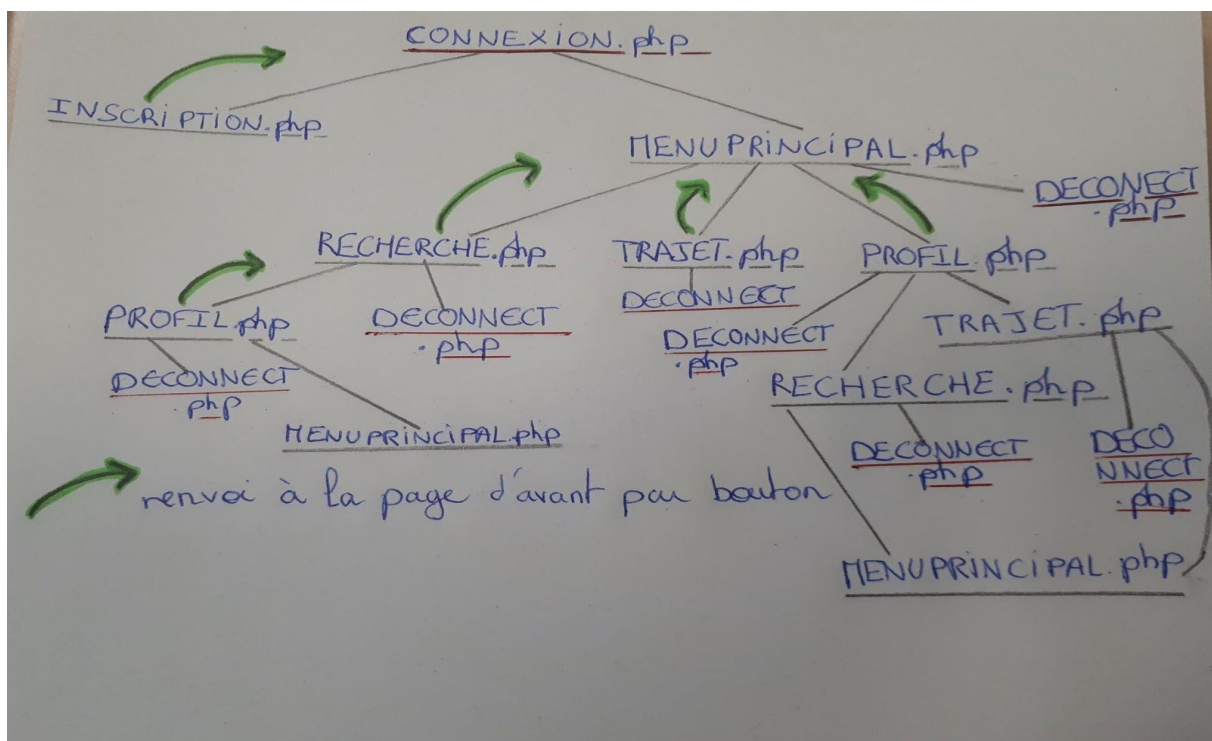


FIGURE 7 : ARCHITECTURE DES PAGE WEB DU SITE

L'arborescence a donc été définie ainsi mais elle sera amenée à évoluer en fonction des options que l'on pourrait envisager de développer en plus des fonctions principales du site.

II. SOLUTIONS PROPOSÉS

Après avoir analysé les besoins et les contraintes du projet nous avons pu définir une potentielle arborescence du site. Voici maintenant le détail des fonctions et des pages qui répondront aux besoins du projet tout en respectant les contraintes et l'arborescence.

1. Connexion et inscription

La fonction de connexion est la première à être utilisée en arrivant sur le site. Si l'utilisateur n'a pas de compte il peut en créer un avec la fonction d'inscription. Les croquis présentés ci-dessous peuvent être modifiés par la suite.

Cette première page présentera donc un module de connexion demandant à l'utilisateur d'entrer son identifiant qui sera une adresse mail ainsi que son mot de passe. Les informations entrées par l'utilisateur seront ensuite comparées aux identifiants enregistrés dans la base de données à l'aide d'une fonction PHP. Si aucun identifiant ne correspond alors l'utilisateur devra recommencer. Sinon il accèdera au menu principal.

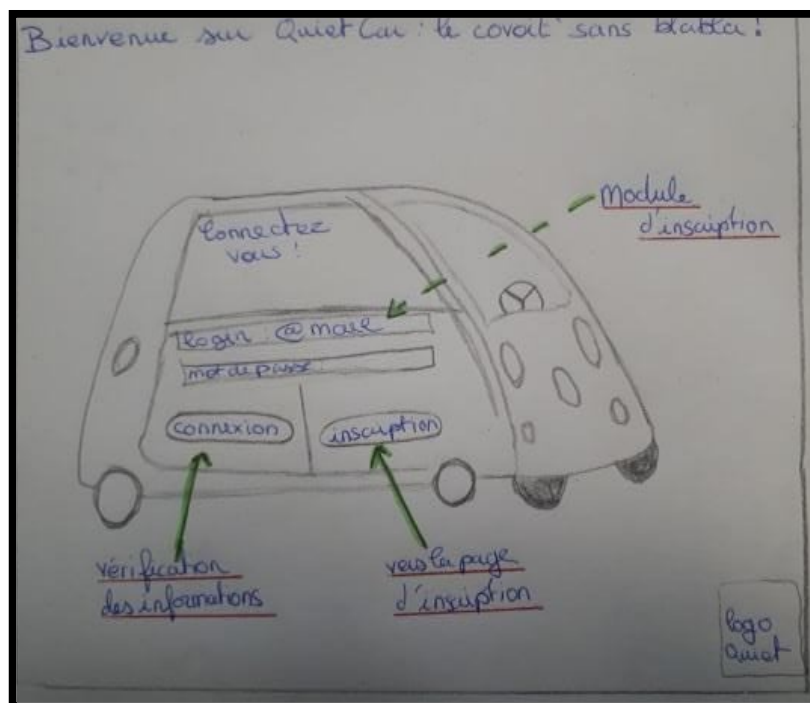


FIGURE 8 : SCHÉMA DE LA PAGE DE CONNEXION

Si l'utilisateur n'a pas de compte, alors il clique sur inscription. Il accède ensuite à un formulaire lui demandant d'entrer les informations suivantes : adresse mail

(identifiant), mot de passe, nom, prénom, genre, numéro de téléphone, description de la personnalité (optionnel). Une fois les informations entrées par l'utilisateur, elles seront stockées dans la base de données et l'utilisateur sera renvoyé vers la page de connexion où en entrant ses nouveaux identifiants il pourra s'authentifier.

Chaque information renseignée par l'utilisateur lors de son inscription va être affichée sur son profil. Les informations pourront être modifiées par la suite.

Le profil présentera les informations suivantes : trajets proposés, numéro de téléphone, personnalité, nombre de trajets effectués en tant que conducteur puis en tant que demandeur de covoiturage, moyenne des notes données pour les anciens trajets effectués en tant que conducteur. Il sera aussi possible d'ajouter une photo de profil.

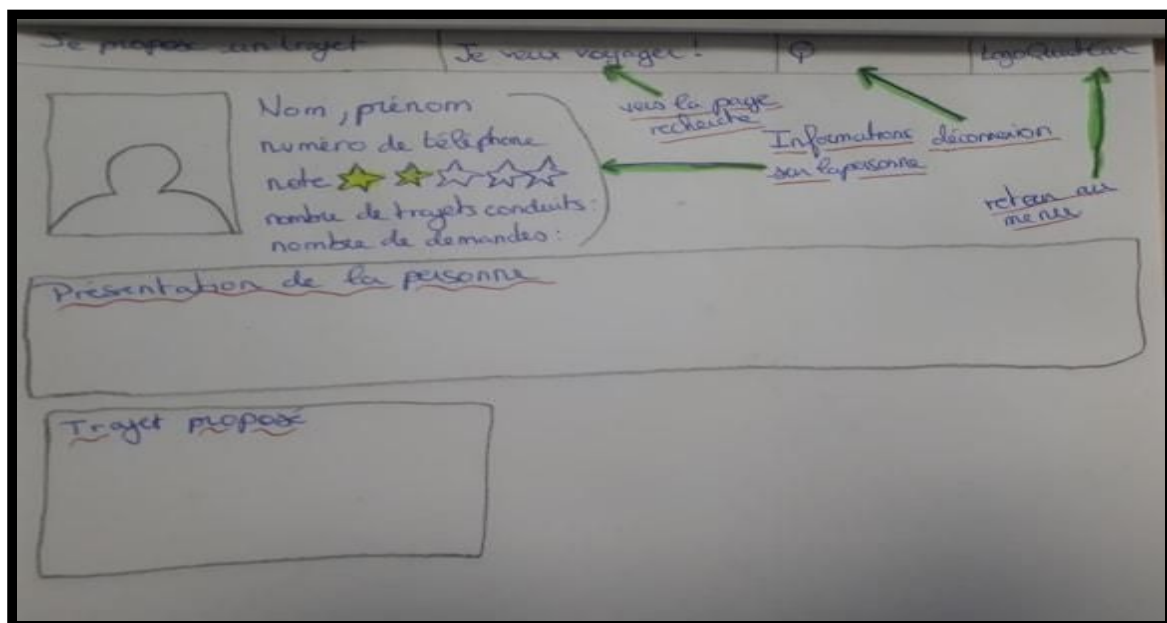


FIGURE 9 : SCHÉMA DE LA PAGE DE PROFIL

2. Les trajets

Deux fonctions vont concerner les trajets : l'ajout de trajets et la recherche de trajets. Ces deux fonctionnalités seront disponibles sur le menu principal et depuis le profil de l'utilisateur.

Le conducteur qui souhaite proposer un trajet se rend sur la page trajet. Il devra alors entrer plusieurs informations.

Tout d'abord, la ville d'arrivée, la ville de départ, les horaires et la date puis les places disponibles qui seront décomptées. Il peut ensuite ajouter d'autres informations sur le

trajet comme sa durée et le nombre de kilomètres mais aussi si le conducteur souhaite prendre l'autoroute.

Enfin, l'utilisateur peut ajouter des informations concernant les passagers en cochant les boutons radio comme, le nombre de bagages, le choix d'accepter les animaux ou pas et si le trajet est non-fumeur. Une fois c'est informations renseignées elles sont ensuite enregistrées dans la base de données et présentées à l'utilisateur.

The image shows a hand-drawn schematic of a web page for proposing a carpooling trip. At the top, there is a small drawing of a car. Below it, the text "Je propose un covoiturage !" is written. The form contains several input fields: "ville de départ", "ville d'arrivée", "J/M/A", "nombre de passagers", "heure de départ", and "voiture". There are also checkboxes for "non-fumeur", "pas de gros bagages", and "pas d'animaux". A text area is labeled "Ajouter une description ou une option". At the bottom, there is a button labeled "Je propose !". Green arrows indicate the flow of the form: from the top right to "déconnexion" and "retour menu", from "J/M/A" to "gère un calendrier", from "voiture" to "type de voiture", and from "Je propose !" to "envoie des infos".

FIGURE 10 : SCHÉMA DE LA PAGE POUR PROPOSER UN TRAJET

Depuis le menu principal ou depuis son profil, l'utilisateur peut rechercher un covoiturage en renseignant une ville de départ et une ville d'arrivée. Il peut aussi renseigner des horaires pour affiner sa recherche. La fonction de recherche va alors consulter sa base de données pour voir si un trajet correspondant à la demande a précédemment été proposé. Il s'affiche alors la liste de tous les covoiturages correspondants. Le passager peut donc faire son choix. Il peut également se rendre sur le profil du conducteur pour consulter ses informations. Une fois le trajet choisi la place est réservé et le nombre de places disponibles sur le covoiturage est décrémenté.

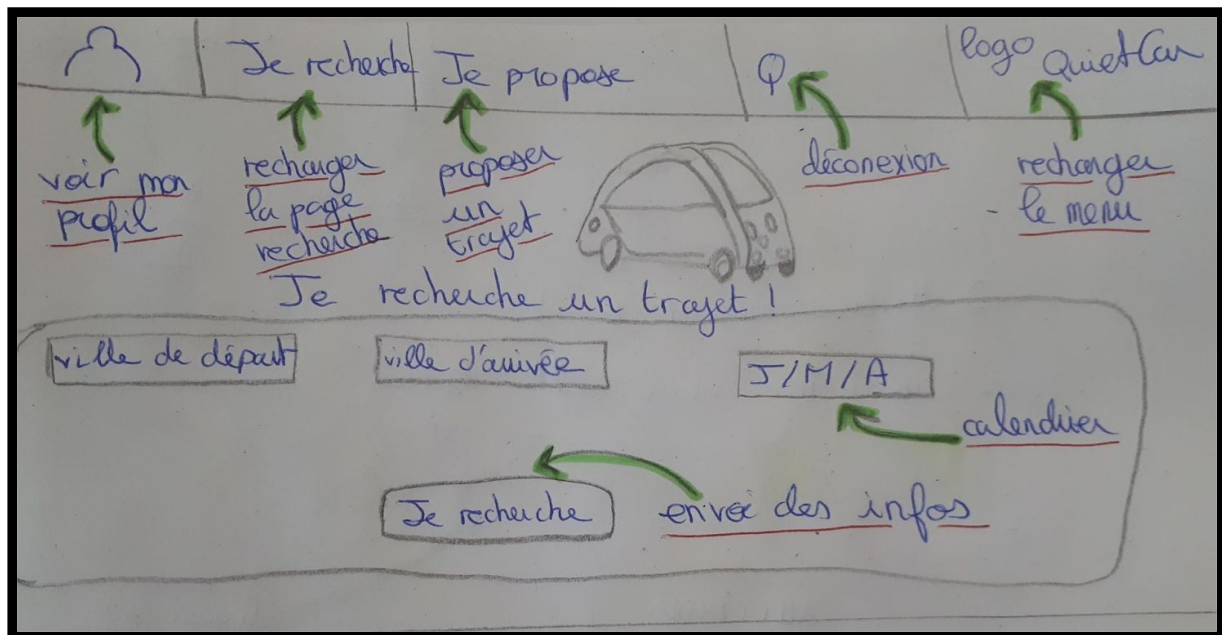


FIGURE 11 : SCHÉMA DE PAGE DE RECHERCHE DE TRAJET

3. Système de paiements

Le client souhaite un système de paiement non monétaire. Nous proposons donc une solution d'attribution d'étoiles par le passager après le trajet.

Une fois le trajet effectué, le passager devra attribuer de 0 à 5 étoiles au conducteur. Le passager pourra également s'il le souhaite, laisser un commentaire sur le profil du conducteur.

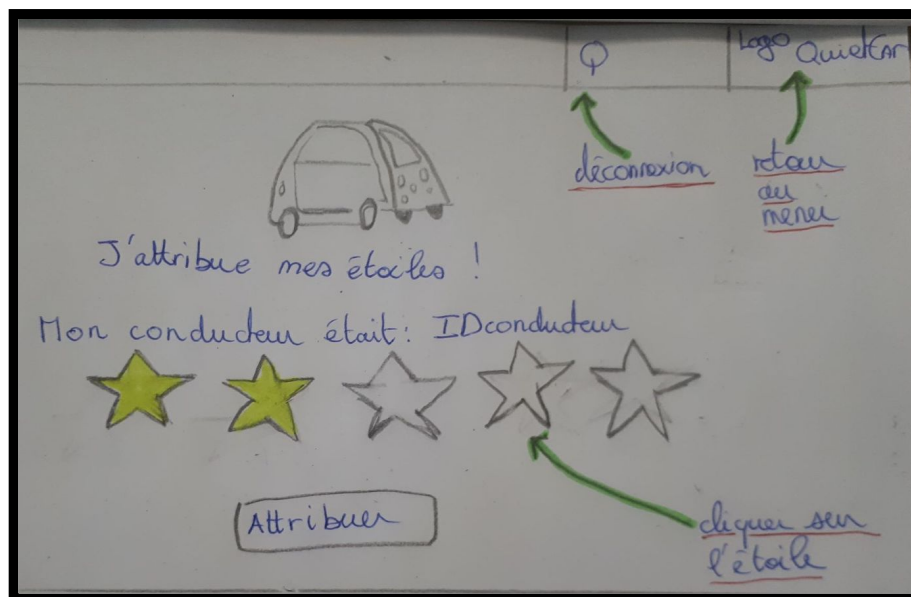


FIGURE 12 : SCHÉMA DE PAGE DE NOTATION D'UN TRAJET

Une moyenne du nombre d'étoiles reçues par le conducteur sera calculée et affichée sur son profil pour conseiller ou non aux autres covoitureurs d'effectuer des trajets avec la personne.

III. BASE DE DONNÉES

Pour réaliser les pages et les fonctions ci-dessus nous avons construit un modèle conceptuel de données (MCD) regroupant toutes les informations nécessaires au bon fonctionnement de l'application.

1. MCD

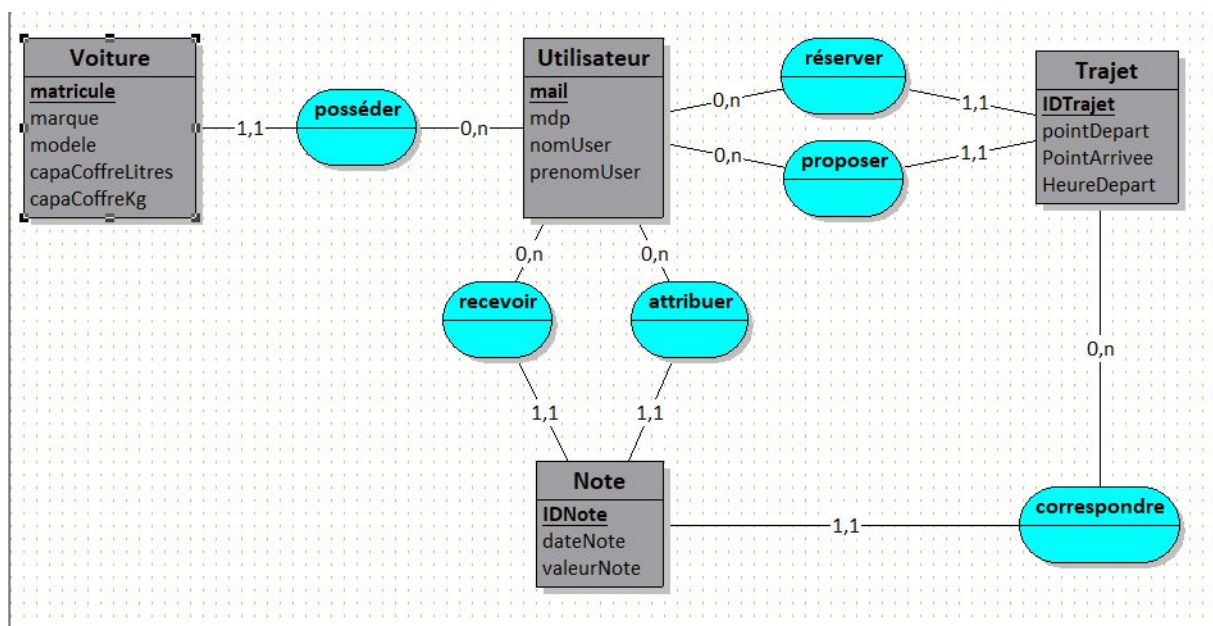


FIGURE 13 : MCD DE QUIETCAR

Dans notre MCD on peut trouver 4 classes d'entités reliées par 6 relations, la classe Utilisateur, le centre de ce MCD, relie toutes les autres classes directement.

L'utilisateur, identifié du système par un ID propre à lui, doit entrer son adresse mail et son mdp pour pouvoir s'authentifier, il est identifiable des autres utilisateurs par son nom et son prenom.

L'utilisateur peut réserver ou proposer des trajets, possèdent un point de départ, un point d'arrivée et une heure de départ.

Une fois effectué, le trajet correspondra à une note attribuée à une certaine date et à une certaine valeur, cette note sera attribuée par un utilisateur à un autre utilisateur.

Enfin, un utilisateur dispose d'une (ou de plusieurs, s'il est un étudiant plutôt riche) voiture identifiée par un matricule, une marque et un modèle, il sera aussi très

important de connaître la capacité du coffre en L et en Kg afin de mieux gérer les bagages transportés.

2. MLD

A partir de notre MCD nous avons pu générer le MLD ci dessous.

Utilisateur = (IDUser, mail, mdp, nomUser, prenomUser);

Trajet = (IDTrajet, pointDepart, PointArrivee, HeureDepart, #IDUser, #IDUser_1);

Voiture = (matricule, marque, modele, capaCoffreLitres, capaCoffreKg, #IDUser);

Note = (IDNote, dateNote, valeurNote, #IDTrajet, #IDUser, #IDUser_1);

En revanche, ce MCD n'est pas encore entièrement compréhensible et pourra être à l'origine de futures confusions concernant la base de données, nous allons donc modifier le nom des clés étrangères de la manière suivante:

Utilisateur=(IDUser, mail, mdp, nomUser, prenomUser);

Trajet=(IDTrajet, pointDepart, PointArrivee, HeureDepart, #IDConducteur,
#IDPassager);

Voiture=(matricule, marque, modele, capaCoffreLitres, capaCoffreKg,
#IDProprietaire);

Note=(IDNote, dateNote, valeurNote, #IDTrajet, #IDUserSource, #IDUserDestination);

CONCLUSION

Après avoir analysé les besoins du client et les contraintes que cela engendre nous avons pu définir l'architecture du site ainsi que ses principales fonctions que nous avons ensuite présentées. Tout cela nous a permis de représenter les bases de données nécessaires au bon fonctionnement du site.

Nous proposons donc une application Web présentant un module d'inscription et de connexion. Les utilisateurs auront tous un profil et pourront proposer des trajets ou bien en rechercher. Le conducteur sera payé en étoiles qui seront affichées sur son profil.

Pour le bon fonctionnement de l'application nous avons réalisé une bases de données avec quatre classes d'entités : voiture, utilisateur, trajet et note .

Ceci était la première version du cahier des charges. Les fonctionnalités les pages et la base de données qui sont présentées seront amenées à être modifiées et améliorées. Une charte graphique sera aussi définie.

TABLE DES ILLUSTRATIONS

Figure 1 : logo BlaBlaCar	6
Figure 2 : logo PostgreSQL	8
Figure 3 : logo PHP	8
Figure 4 : logo JavaScript	8
Figure 5 : Logo W3C	9
Figure 6 : logo Bootstrap	9
Figure 7 : architecture des page web du site	10
Figure 8 : schéma de la page de connexion	11
Figure 9 : schéma de la page de profil	12
Figure 10 : schéma de la page pour proposer un trajet	13
Figure 11 : schéma de page de recherche de trajet	14
Figure 12 : schéma de page de notation d'un trajet	14
Figure 13 : MCD de QuietCar	16